



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV MIKROELEKTRONIKY

DEPARTMENT OF MICROELECTRONICS

NÁVRH VYBRANÉ ČÁSTI STANDARDU IEEE 802.1Q

DESIGN OF SELECTED IEEE 802.1Q STANDARD PARTS

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. Filip Kliment

VEDOUCÍ PRÁCE

SUPERVISOR

doc. Ing. Lukáš Fajcik, Ph.D.

BRNO 2018

Diplomová práce

magisterský navazující studijní obor **Mikroelektronika**
ÚSTAV MIKROELEKTRONIKY

Student: Bc Filip Kliment

ID: 164875

Ročník: 2

Akademický rok: 2017/18

NÁZEV TÉMATU:

NÁVRH VYBRANÉ ČÁSTI STANDARDU IEEE 802.1Q

POKYNY PRO VYPRACOVÁNÍ:

Cílem diplomové je vytvořit návrh funkčního IP jádra standardu IEEE 802.1QBU v jazyku VHDL. Cílem dále bude úprava MAC vrstvy pro podporu preempce ethernetových rámců popsané ve standardu IEEE 802.3BR. Celé jádro bude používat rozhraní Avalon Streaming (rozhraní podporované Intel FPGA). IP jádro bude testované simulací nástrojem Modelsim s použitím vlastních verifikačních BFM

DOPORUČENÁ LITERATURA:

Podle pokynů vedoucího práce

Termín zadání: 18.9.2017

Termín odevzdání: 14.12.2017

Vedoucí práce: doc. Ing. Lukáš Fucík, Ph.D.

Konzultant:

doc. Ing. Lukáš Fucík, Ph.D.
předseda oborové rady

UPOZORNĚNÍ:

Autor semestrální práce nesmí při vytváření semestrální práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

ABSTRACT

This thesis deals with network substandards from the TSN group (IEEE 802.1Q), which deal with prioritization of network traffic in TSN networks. These sub-standards include 802.1QBV and 802.1QBU, which have been described in more detail and compared in terms of network permeability and latency. Substandard 802.1QBU was chosen for the design implementation in FPGA. The design was described in VHDL. The designed design was verified by simulations, using self-tests. The work includes synthesis and time analysis.

KEYWORDS

Ethernet, TSN, IEEE, OSI, MAC, Frame preemption, Data Link Layer, FPGA, VHDL

ABSTRAKT

Táto práca sa zaoberá sieťovými podštandardmi z kategórie TSN (IEEE 802.1Q), ktoré riešia prioritizáciu sieťovej prevádzky v TSN sieťach. Medzi tieto podštandardy patria 802.1QBV a 802.1QBU, ktoré boli detailnejšie popísané a porovnané z hľadiska priepustnosti a latencie. Podštandard 802.1QBU bol zvolený pre návrh implementácie do obvodu FPGA. Dizajn bol popísaný v jazyku VHDL. Navrhnutý dizajn bol overený simuláciami, s využitím samočinných testov. V práci sú zahrnuté aj výsledky syntézy a časovej analýzy.

KLÍČOVÁ SLOVA

Ethernet, TSN, IEEE, OSI, MAC, Frame preemption, Data Link Layer, FPGA, VHDL

KLIMENT, F. NÁVRH VYBRANÉ ČÁSTI STANDARDU IEEE 802.1Q . Brno:
Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií,
2018. 55 s. Vedoucí diplomové práce doc. Ing. Lukáš Fojcik, Ph.D..

Prehlásenie

Prehlasujem, že svoju diplomovú (bakalársku) prácu na Návrh vybrané časti standardu IEEE 802.1Q som vypracoval samostatne pod vedením vedúcej / ho diplomovej (bakalárskej) práce as použitím odbornej literatúry a ďalších informačných zdrojov, ktoré sú všetky citované v práci a uvedené v zozname literatúry na konci práce . Ako autor uvedenej diplomovej (bakalárskej) práce ďalej prehlasujem, že v súvislosti s vytvorením tejto diplomovej (bakalárskej) práce som neporušil autorské práva tretích osôb, najmä som nezasiahol nedovoleným spôsobom do cudzích autorských práv osobnostných a som si plne vedomý následkov porušenia ustanovenia § 11 a nasledujúcich autorského zákona č. 121/2000 Zb., vrátane možných trestnoprávných dôsledkov vyplývajúcich z ustanovenia časti druhej, hlavy VI. diel 4 Trestného zákonníka č. 40/2009 Zb.

V Brně dne: **22. května 2018**

.....
podpis autora

Pod'kování

Ďakujem vedúcemu diplomovej práce doc. Ing. Lukáš Fucík, Ph.D. a konzultantovi diplomovej práce Ing. Petr Grillinger, Ph.D. za účinnú metodickú, pedagogickú a odbornú pomoc a ďalšie cenné rady pri spracovaní mojej diplomovej práce. Ďalej chcem poďakovať spoločnosti TTech za poskytnutie prostriedkov k vypracovaniu tejto diplomovej práce.

V Brně dne: **22. května 2018**

.....
podpis autora

Experimentální část této diplomové práce byla realizována na
výzkumné infrastruktuře vybudované v rámci projektu
CZ.1.05/2.1.00/03.0072

**Centrum senzorických, informačních a komunikačních systémů
(SIX)**

operačního programu Výzkum a vývoj pro inovace.

OBSAH

Zoznam symbolov a zkratiek	9
Zoznam obrázkov	10
Zoznam tabuliek	11
Úvod	12
1 Model osi (Open systems interconnection model)	13
1.1 Fyzická vrstva	13
1.2 Linková vrstva	13
1.2.1 Logická linkovacia vrstva LLC (Logical Link Control sublayer)	13
1.2.2 Media Access Control sublayer (MAC)	14
1.3 Efektívnosť prenosu (dátové komunikácie a vytváranie sietí)	14
2 802.1QBV	17
2.1 Funkcie podporované 802.1QBV	18
2.1.1 Vytváranie chráneného okna, používaním operácií na bránach	18
2.1.2 Scheduled traffic and end stations	19
2.1.3 Plánovná prevádzka a koncové stanice.....	19
2.2 Prepojenie vo vnútri 802.1QBV	20
3 802.1QBU	21
3.1 Funkcie 802.1QBU	22
3.1.1 Preemptcia použitá v izolácii.....	22
3.1.2 Plánovanie a preemptcia použitá v kombinácii bez HOLD/RELEASE	22
3.1.3 Plánovanie a preemptcia použitá v kombinácii s HOLD/RELEASE ..	22
3.1.4 Pridelovanie šírky pásma a expresná prevádzka siete.....	23
3.2 Prepojenie v 802.1QBU	23
4 802.1QBU Špecifikácia	25
4.1 Funkcie pre 802.1QBU	25
4.1.1 Fronty pre ethernet rámce	25
4.1.2 Výber prenosu.....	25
4.1.3 Algoritmus pre tvarovač založený na kreditovom systéme	26

4.1.4	Vylepšený algoritmus pre výber prenosu	27
4.1.5	Preempcia ethernet rámcov	27
4.2	MAC service	28
4.3	Parametre potrebné pre funkciu preempciu ethernet rámcov	28
4.3.1	framePreemptionStatusTable štruktúry a dátové typy	28
4.3.2	Objekt s holdAdvance	28
4.3.3	Objekt s releaseAdvance	28
4.3.4	Objekt s preemptionActive	28
4.3.5	Objekt s holdRequest	29
5	802.3BR	30
5.1	Formát rámcu podľa 802.3BR	30
5.2	Kódovanie SMD a čísla fragmentu	31
6	Teoretický výpočet a porovnanie priepustnosti 802.QBU s inými štandardmi	32
6.1	Porovnanie priemernej sieťovej prevádzky cez 802.1QBU a cez štandardný ethernet	32
6.2	Porovnanie uvažovanej sieťovej prevádzky cez ethernet, 802.1QBU, 802.1QBV a ich kombinácii	33
6.3	Porovnanie latencie	36
7	Implementácia podštandardu 802.1QBU	38
7.1	RX MAC	39
7.2	TX MAC	41
7.3	CRC_REMOVER	45
7.4	Syntéza	45
8	Simulácie	46
8.1	Test bench tb_top_rx	46
8.1.1	Test procedúra tp_preamble	47
8.1.2	Test procedúra tp_express	47
8.1.3	Test procedúra tp_no_preemption	48
8.1.4	Test procedúra tp_preemption	48
8.2	Test bench tb_top_final	49
8.2.1	Test procedúra tp_padding	51
8.2.2	Test procedúra tp_back_to_back	51
8.2.3	Test procedúra tp_long_pckt	51

8.3	Výsledky test procedúr	52
9	Záver	53
	Literatúra	55

ZOZNAM SYMBOLOV A ZKRATIEK

ALUT	Adaptívna Look Up tabuľka
CRC	Cylická redundačná kontrola
eMAC	expresný MAC
ETS	Vylepšený výber vysielania
FCS	kontrolná sekvencia rámcu
FIFO	prvý dnu, prvý von
FPGA	Programovateľné hradlové pole (Field-programmable gate array)
IP	Duševné vlastníctvo (Intellectual Property)
IEEE	Inštitút pre elektrotechnické a elektronické inžinierstvo
LAN	Lokálna sieť
LLC	Logical Link Control
MAC	Media Access Control
MFCS	kontrolná sekvencia pre Mframe
MII	Médiovo nezávislé rozhranie
OSI	Open Systems Interconnection
PFC	Priority-based Flow Control
PHY	Fyzická vrstvar
pMAC	preempčný MAC
PTP	Precízny časový protokol
SFD	Start Frame Delimiter
SMD	Start Mframe Delimiter
SMD-Cx	Start Mframe Delimiter Not Initial (Continue) Fragment
SMD-Ix	Start Mframe Delimiter Initial Fragment
TSN	Časovo citlivé siete
VHDL	VHSIC Hardware Description Language
VLAN	Virtuálna LAN

ZOZNAM OBRÁZKOV

Obr. 1.1: Graf závislosti priepustnosti na veľkosti rámcu. [14]	15
Obr. 2.1: Ochranné pásmo A) pevný začiatok ochranného pásma B) premenný začiatok ochranného pásma [7].	18
Obr. 2.2: Príklad operácií prebiehajúcich na bránach.....	19
Obr. 2.3: Schematické zobrazenie zapojenia pre podštandard QBVčČ.	20
Obr. 3.1: Príklad preempcie rámcov.	21
Obr. 3.2: Bloková schema pre 802.1QBU. [12]	24
Obr. 5.1: Porovnanie formátov klasického MAC rámca, Expresného rámca a Preemptovateľného rámca. [11].....	30
Obr. 5.2: Formáty fragmentov preemptovateľného rámcu. [11]	31
Obr. 6.1: Závislosť priemernej priepustnosti na dĺžke rámca, pre štandardný ethernet a varianty podštandardu s rôznou minimálnou dĺžkou fragmentu.....	33
Obr. 6.2: Porovnanie závislostí priepustností pre štandardný ethernet QBV, QBU s minimálnou dĺžkou fragmentu 64 bajtov a kombinácie QBV s QBU s minimálnou dĺžkou fragment 64 bajtov.	34
Obr. 6.3: Porovnanie závislostí priepustností pre štandardný ethernet QBV, QBU s minimálnou dĺžkou fragmentu 128 bajtov a kombinácie QBV s QBU s minimálnou dĺžkou fragment 128 bajtov.	35
Obr. 6.4: Porovnanie závislostí priepustností pre štandardný ethernet QBV, QBU s minimálnou dĺžkou fragmentu 256 bajtov a kombinácie QBV s QBU s minimálnou dĺžkou fragment 256 bajtov.	36
Obr. 6.5: Porovnanie veľkosti latencie pre expresné rámce.	37
Obr. 7.1: Top modul implementovaného dizajnu, s prepojením vnútorných komponentov.	39
Obr. 7.2: Diagram stavového automatu v RX MAC.	41
Obr. 7.3: Diagram stavového automatu v TX MAC.....	44
Obr. 8.1: Bloková schéma test bencu tb_top_rx.	46
Obr. 8.2: Bloková schéma test bencu tb_top_final.	50

ZOZNAM TABULIEK

Tabuľka 4-1: Doporučené mapovanie priorít pre triedy sieťovej prevádzky. [10]	25
Tabuľka 4-2: Identifikátori algoritmu pre výber prenosu. [10]	26
Tabuľka 5-1: Hodnty pre SMD polia a polia s číslom fragmentu. [11].....	31
Tabuľka 7-1: Tabuľka prechodov medzi stavmi pre automat v TX MAC.	40
Tabuľka 7-2: Tabuľka výstupov pre stavový automat v RX MAC.	41
Tabuľka 7-3: Tabuľka prechodov medzi stavmi pre automat v TX MAC.	42
Tabuľka 7-4: Tabuľka výstupov pre stavový automat v TX MAC.	43
Tabuľka 7-5: Prehľad výsledkov syntézy.	45

ÚVOD

V súčasnosti je pozornosť venovaná vývoju protokolov IEEE pod názvom TSN (Time-Sensitive Networking). V praxi sa už používa veľké množstvo protokolov zo skupiny TSN, ale mnohé z nich sú stále vo vývoji a nie sú implementované v komerčných zariadeniach. Protokoly skupiny TSN zaisťujú deterministický prenos dát prostredníctvom sietí IEEE 802. Toto je umožnené protokolmi zaisťujúcim dopravu s nízkou latenciou, časovou determinističnosťou správ a s predchádzaním strát paketov.

Základné teoretické znalosti zo sietí, na ktorých je postavený cieľ tejto diplomovej práce, sú popísané v kapitole jedna.

Hlavným cieľom tejto diplomovej práce je analyzovať niektoré z podštandardov IEEE 802.1Q a vybrať si jeden z nich, pre implementáciu v jazyku VHDL. V tejto práci sú hlbšie analyzované dva z nich, pretože sa predpokladá ich spolupráca, podľa štandardu 802.1Q, v riešeníach pre TSN. Prvým z analyzovaných podštandardov je 802.1QBV, vylepšené plánovanie sieťovej prevádzky. Ďalší analyzovaný podštandard je 802.1QBU, ktorý slúži k popisu správneho fungovania pri preempcii rámcov. Tieto podštandardy sú opísané v kapitolách dva a tri. Porovnaním ich vlastností sa zaoberá šiesta kapitola. V tejto kapitole sú porovnané teoretické hodnoty dátovej priepustnosti a latencie časovo kritických rámcov spolu s okomentovanými výsledkami.

Pre implementáciu bol zvolený štandardný 802.1QBU. Funkcie, ktoré rozširujú a zlepšujú tento podštandard, sú popísané v štvrtej kapitole spolu s ich parametrami. Kódovaním pri preempcii rámcov sa zaoberá štandard 802.3BR, ten je popísaný v piatej kapitole.

Dizajn navrhovaného IP jadra je popísaný v siedmej kapitole. Sú v nej popísané funkcie, ktoré dizajn vykonáva. Na konci tejto kapitoly sú tiež uvedené výsledky zo syntézy, ktoré podávajú informácie o možnosti nahrania dizajnu do zvoleného FPGA.

Pre overenie správnosti fungovania dizajnu boli navrhnuté test benchy, ktoré sú riadené test procedúrami. Tieto test benchy a test procedúry sú detailne popísané v ôsmej kapitole. Je v nej možné nájsť schematické a funkčné zapojenie vo vnútri test benchu. Pre každý test bench sú v nasledujúcich podkapitolách popísané test procedúry, ktoré ovládajú test bench.

1 MODEL OSI (OPEN SYSTEMS INTERCONNECTION MODEL)

Model osi(Open System Interconnection) je nástroj pre lepšie pochopenie dátovej komunikácie medzi dvomi sieťovými systémami. Tento model je rozdelený do siedmych vrstiev. Každá z vrstiev má špecifickú funkciu pre vrstvu nad ňou a ponúka servisné funkcie pre vrstvu pod ňou. Tri najspodnejšie vrstvy sú zamerané na prepúšťanie prevádzku prostredníctvom siete do koncového systému. Vrchné štyri vrstvy sú použité v koncovom systéme pre kompletizovanie komunikačného procesu.

Sieťový model ponúka všeobecné prostriedky na oddelenie funkcií počítačových sietí do viacerých vrstiev. Takýto model vrstvených funkcií je tiež nazývaný "protokolový zásobník" alebo "protokolový balíček".

Protokoly alebo pravidlá môžu fungovať buď v hardvéri, softvéri, alebo rovnako ako pri väčšine protokolových balíčkov v kombinácii týchto dvoch. Charakter týchto kombinácií je, že spodné vrstvy vykonávajú svoju prácu pomocou hardvéru alebo firmvéru (softvér, ktorý beží na špecifickom hardvéri), zatiaľ čo vyššie vrstvy spracováva software.

Tento model je definovaný normou ISO (Medzinárodná organizácia pre normalizáciu) štandard 7498-1. Tento model umožňuje, aby všetky prvky siete fungovali spoločne, bez ohľadu na to, kto vytvoril protokoly a aký počítačový predajca ich podporuje.

1.1 Fyzická vrstva

Fyzická vrstva je umiestnená na spodku modelu OSI. Táto vrstva je navrhnutá na prenos bitových prúdov pomocou elektrických signálov, svetiel, rádiových prenosov. [2] Táto vrstva tiež definuje konektor, špecifikácie rozhrania a požiadavky na médium (kábel). Elektrické, mechanické, funkčné a procedurálne špecifikácie sú poskytované na odosielanie bitových tokov v počítačovej sieti. [1]

1.2 Linková vrstva

Linková vrstva definuje komunikáciu na úrovni odkazu, ktorá je nezávislá od média. Toto zariadenie je postavené na médiu závislom na fyzickom kanáli, ktorý poskytuje fyzická vrstva [3]. Vrstva dátového spojenia pozostáva z mnohých podvrstiev, z ktorých najdôležitejšie sú Media Access Control (MAC) a Logical Link Control (LLC) a niektoré ďalšie, ktoré vytvárajú lepšie rozhranie, napríklad MAC Control [3].

1.2.1 Logická linkovacia vrstva LLC (Logical Link Control sublayer)

LLC je horná podvrstva vrstvy dátového spojenia, táto podvrstva je potrebná pre multiplexný a demultiplexný mechanizmus, multiplexný mechanizmus zabezpečuje

koexistenciu niektorých sieťových protokolov v multibodovej sieti a robí prenos týchto 10 protokolov na rovnakom médiu, ďalšie mechanizmy a procesy poskytované LLC sú riadenie prietoku a požiadavka na automatické opakovanie. LLC vytvára rozhranie medzi MAC a sieťovou vrstvou. [4]

1.2.2 Media Access Control sublayer (MAC)

MAC prijíma a prenáša kompatibilné rámce. Vytvára a riadi sekvenciu kontroly rámcov (FCS), ktorá je vytvorená a riadená cyklickou kontrolou redundancie(CRC). Zahadzuje deformované rámce. Taktiež pridáva a odstraňuje preambulu a oddeľovač začiatku rámcu (SFD). [5]

Existujú tri typy ethernetového rámcu, ale všetky majú rovnaký formát, len sú rôznej dĺžky:

- Každý rámec začína s preambulou, sedem byteových polí, ktoré majú rovnakú hodnotu, 0x55, platí to pre všetky typy rámcu[5]
- SFD je nasledujúci byte v rámci, pri štandardných rámcach má vždy hodnotu 0xD5. [5]
- Hlavička rámcu sa skladá zo dvoch šesť byteových polí, MAC cieľová adresa a MAC zdrojová adresa, ďalej tam môže byť pridané štvor byteové pole štítok(tag), EtherType je ďalšia časť dvojbyteová časť rámcu. Používa sa na označenie toho, ktorý protokol je zapuzdrený v užitočných dátach rámcu. Rovnaké pole sa tiež môže použiť aj na označenie veľkosti niektorých ethernetových rámcov.[5]
- Užitočné zaťaženie je pole vyplnené užitočnými dátami, jeho dĺžka je od 42/46 oktetov po 1500 oktetov, ak je menej oktetov ako minimum, MAC pridá zostávajúce byty, nazývané výplň, tieto byty majú vždy hodnotu 0x00. [5]
- FCS je štvor byteové pole, ktoré pomáha zisťovať poškodenie pomocou výpočtu CRC. CRC sa vypočítava zo zdrojovej a cieľovej adresy, poľa dĺžky / typu a užitočného zaťaženia. [5]

1.3 Efektívnosť prenosu (dátové komunikácie a vytváranie sietí)

Jedným z cieľov dátovej komunikačnej siete je presunúť čo najvyšší objem presných informácií prostredníctvom siete. Čím vyšší je objem dát, tým väčšia je výsledná efektívnosť siete. Účinnosť siete je ovplyvnená charakteristikami obvodov, ako sú chybovosť a maximálna prenosová rýchlosť, rovnako ako rýchlosť pre vysielanie a príjem zariadení, ich detekcia chýb a kontrolné metódy, a protokoly používané na linkovacej vrstve.[14]

Každý protokol používa niektoré bity alebo bajty na vymedzenie začiatku a konca každej správy a na kontrolu chýb. Tieto bity a bajty sú nevyhnutné na vysielanie, ale nie sú súčasťou správy. Nedodávajú používateľovi informačnú hodnotu, ale počítajú sa proti celkovému počtu bitov, ktoré sa môžu prenášať. [14]

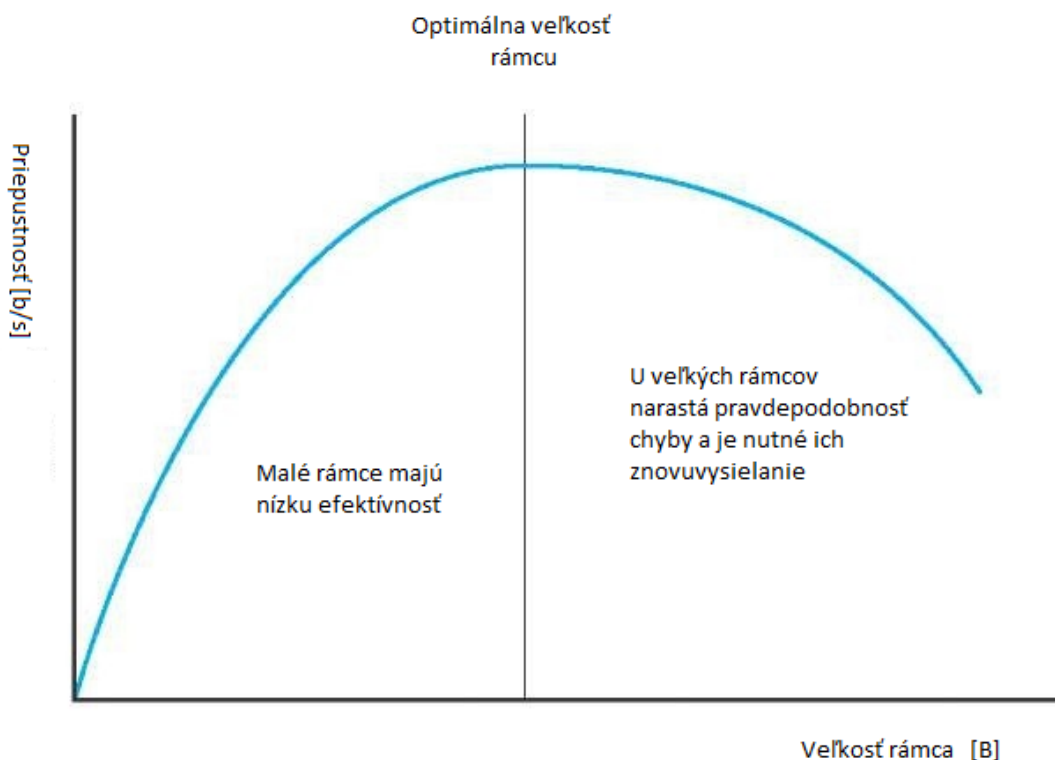
Každý komunikačný protokol má informačné bity a prídavné bity. Informačné bity sú tie, ktoré slúžia na vyjadrenie informácie. Prídavné bity sa používajú na účely ako je kontrola chýb a označenie začiatku a konca znakov a paketov. Bit parity, ktorý sa používa na kontrolu chýb, je prídavný bit, pretože sa nepoužíva na prenos informácií. [14]

Efektívnosť prenosu je definovaná ako podiel celkového počtu bitov informácií (bity odoslané v správe) s celkovým počtom bitov v prenosu (informačné bity plus prídavné bity). [14]

Rovnaký základný vzorec sa môže použiť na výpočet účinnosti synchronného prenosu. [14]

Všeobecným pravidlom je, že čím väčšie je pole správy, tým je protokol efektívnejší, ale ak sa kedykoľvek prijme rámec obsahujúci chybu, celý rámec sa zahodí a musí byť znovu vyslaný, preto sa nepoužívajú štandardné rámce väčšie ako 1536 bajtov. Pravdepodobnosť, že rámec obsahuje chybu, sa zvyšuje s veľkosťou rámca, väčšie rámce pravdepodobnejšie obsahujú chyby. Menšie rámce sa používajú kvôli zákonom pravdepodobnosti. [14]

Na Obr.1.1 je zobrazený graf závislosti priepustnosti dátovej komunikácii na dĺžke vysielaného rámca. [14]



Obr. 1.1: Graf závislosti priepustnosti na veľkosti rámca. [14]

Priepustnosť je celkový počet bitov informácií prijatých za sekundu po zohľadnení pridaných bitov a potreby opakovaného vysielania rámcov obsahujúcich chyby.

Všeobecne povedané, malé rámce poskytujú lepšiu priepustnosť pre komunikáciu s viacerými chybami, zatiaľ čo väčšie rámce poskytujú lepšiu priepustnosť v sieťach menej náchylných na chyby. Našťastie vo väčšine reálnych sietí je krivka znázornená na obr.1.1 na vrchu veľmi plochá, čo znamená, že existuje veľa rozmerov rámcov, ktoré poskytujú takmer optimálny výkon. Veľkosti rámcov sa medzi jednotlivými sieťami značne líšia, ale ideálna veľkosť rámu sa pohybuje medzi 2000 a 10 000 bajtami. [14]

Optimalizácia výkonu v sieti, najmä v sieti klient-server, môže byť zložitá, pretože význam veľkosti rámca vie veľmi ovplyvniť rýchlosť komunikácie. Výber správnej alebo nesprávnej veľkosti rámcu môže mať väčší vplyv na výkon ako čokoľvek, čo by ste mohli robiť na serveri. [14]

Ideálna veľkosť rámcu závisí od konkrétnej aplikácie a vzoru správ, ktoré generuje. Pre štandardnú komerčnú veľkosť je ideálna veľkosť rámcu pohybujúca sa medzi 4000 a 8000 bajtami. Bohužiaľ, nie všetky sieťové softvérové balíky umožňujú manažérom siete jemne doladiť veľkosti rámcu týmto spôsobom. Ale štandardizovaná veľkosť rámcu je iba 1536 bajtov a to z toho dôvodu, že Ethernet bol štandardizovaný mnoho rokov dozadu, vtedy bola pravdepodobnosť chyby vyššia. Jumbo a super Jumbo rámce vznikli na optických sieťach s vysokou rýchlosťou prenosu a veľmi nízkou pravdepodobnosťou výskytu chyby. [14]

Výpočet skutočnej výkonnosti dátovej komunikačnej siete je zložitý, pretože závisí nielen od efektívnosti protokolu dátového spojenia, ale aj od miery chybovosti a počtu opakovaných prenosov. Rýchlosť vysielania informačných bitov (TRIB) je mierou účinného počtu informačných bitov, ktoré sa prenášajú cez komunikačný obvod za jednotku času. Základná rovnica TRIB z ANSI je definovaná vzťahom číslo 1.1. [14]

$$TRIB = \frac{K(M - C)(1 - P)}{\left(\frac{M}{R}\right) + T} \quad (1.1)$$

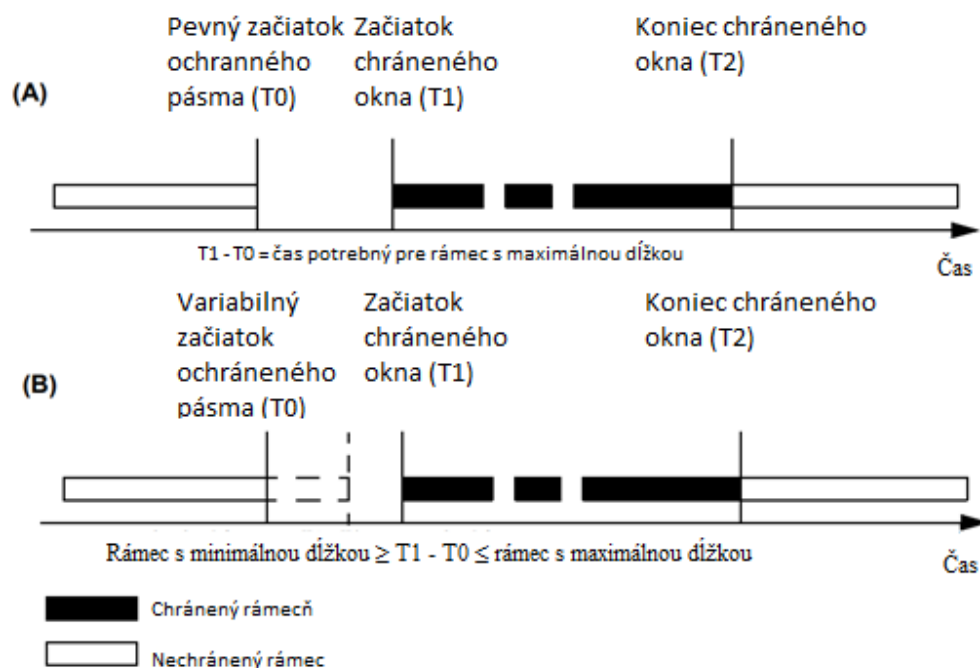
Kde:	K	je počet informačných bitov na jeden znak
	M	je dĺžka rámcu (počet znakov v rámci)
	R	je rýchlosť prenosu údajov v počte znakov za sekundu
	C	je priemerný počet neinformačných znakov na jeden rámec
	P	je pravdepodobnosť, že rámec bude potrebovať znovu odoslanie z dôvodu výskytu chyby
	T	je čas medzi rámcami v sekundách

2 802.1QBV

Štandard pre lokálne a metropolitné siete - mosty a virtuálne premost'ované lokálne siete Modifikácia: zlepšenia plánovanej prevádzky. [6]

V tomto dodatku sú špecifikované procedúry na vyprázdnovanie radov podľa časovania rozmeru. Plánovanie prenosu rámcov na základe časovania odvodeného zo štandardu IEEE 802.1AS, ktoré je povolené riadenými objektmi a rozšíreniami existujúcich protokolov. Štítky virtuálnej lokálnej siete (VLAN) sú zakódované hodnoty priority umožňujúce simultálnu naplánovanú prevádzku, kreditovo založené tvarovanie prevádzky a iný premostený prenos cez lokálne siete (LAN). Použitie tohoto dodatku prináša lepšie predvídateľné doručovanie rámcov v čase, v ktorom prenos rámcov začne a celkové oneskorenie s časovým nepokojom, ktoré sa vyskytnú v mieste určenia. Toto umožní zmiešať časovo kritickú prevádzku s inými triedami premávky v tej istej sieti, ak toto miešanie spĺňa časové požiadavky časovo kritickej premávky. Urobenie priorít je nedostačujúce, preto by malo byť zabezpečené, že rámec s nižšou prioritou majú dostatok času na dokončenie odosielania pred začiatkom času vyhradeného pre prenos kritického rámcu, inak by mohlo dôjsť k oneskoreniu vysielania až o čas potrebný pre prenos rámcu s maximálnou dĺžkou. [7]

Pre implementáciu je potrebné zabezpečiť, aby prístup k vysielacím rámcom mal iba jedna komunikačná trieda alebo skupina tried, toto vytvára chránený kanál pre túto triedu, iný prenos musí byť zastavený s dostatočnou časovou rezervou pred začiatok chráneného kanálu. Posledný prenos nechránených kanálov musí začať minimálne čas potrebný na prenos rámcu s maximálnou dĺžkou pred časom začiatku chráneného okna. Toto vytvára ochranné pásmo pred časom určeným pre chránenú prevádzku. V tomto ochrannom pásme nemôže začať vysielanie žiadneho rámcu. Šírka ochranného pásma môže byť nastavená na časovú dĺžku potrebnú na prenos rámcu s maximálnou dĺžkou, alebo učením veľkosti ďalšieho rámcu v rade na prenos a ak je dostatok času na prenos rámcu s touto dĺžkou, je možné začať prenos tohoto rámcu. Obydve riešenia sú zobrazené na obr. 2.1.[7]



Obr. 2.1: Ochranné pásmo A) pevný začiatok ochranného pásma B) premenný začiatok ochranného pásma [7].

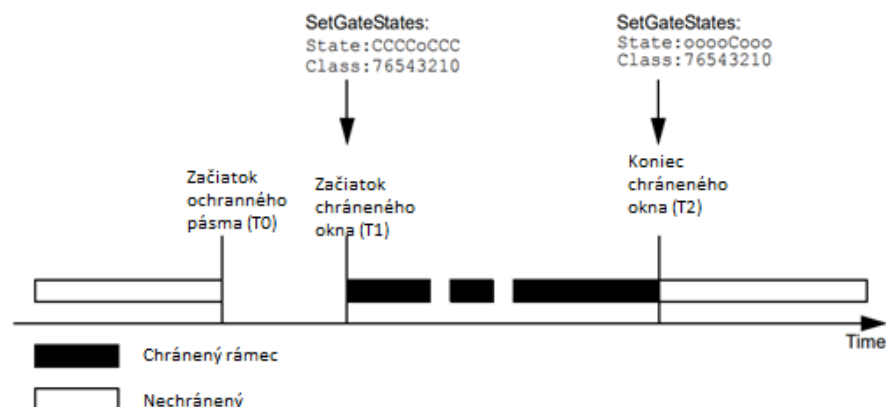
2.1 Funkcie podporované 802.1QBV

V nasledujúcich podkapitolách sú popísané procesy, premenné a funkcie podporované a poskytované podštandardom 802.1QBV.

2.1.1 Vytváranie chráneného okna, používaním operácií na bránach

Vylepšenia implementácie pre plánovanú prevádzku umožňuje zapnutie a vypnutie prenosu načasovanú pre všetky triedy prevádzky na príslušnom porte. Všetky dopravné triedy majú individuálne spínacie prenosové brány, ktoré sú riadené zoznamom operácií pre brány, tento zoznam môže byť programovateľný. Funkcia SetGateStates funguje s parametrom časového oneskorenia, ktorý je časom medzi dvoma po sebe nasledujúcimi operáciami, a parametrom GateState, čo je osem bitový vektor, ktorý definuje stav každej brány pre celé dopravné triedy. Táto operácia môže vytvoriť akúkoľvek kombináciu stavu brán a chránených tried. Ukončí sa na konci zoznamu a reštartuje sa po uplynutí OperCycleTime od začiatku ciklu. Obr 2.2 opisuje, ako môžu

operácie brány vytvoriť chránené okno. V čase T1 sa nastavuje brána pre chránenú dopravnú triedu do stavu otvorená a brány pre ostatné dopravné triedy nastavujú do stavu zatvorená. V čase T2 táto operácia zatvorí bránu pre túto triedu a zároveň otvorí brány ostatným triedam.



Obr. 2.2: Príklad operácií prebiehajúcich na bránach.

2.1.2 Scheduled traffic and end stations

Spustenie protokolu PTP a dostupnosť času PTP umožňujú lepšie načasovanie operácií brány. Tento protokol nie je potrebný na implementáciu, môže byť nahradený inými alternatívnymi mechanizmami, ktoré zabezpečujú spoločný pohľad na čas. [7]

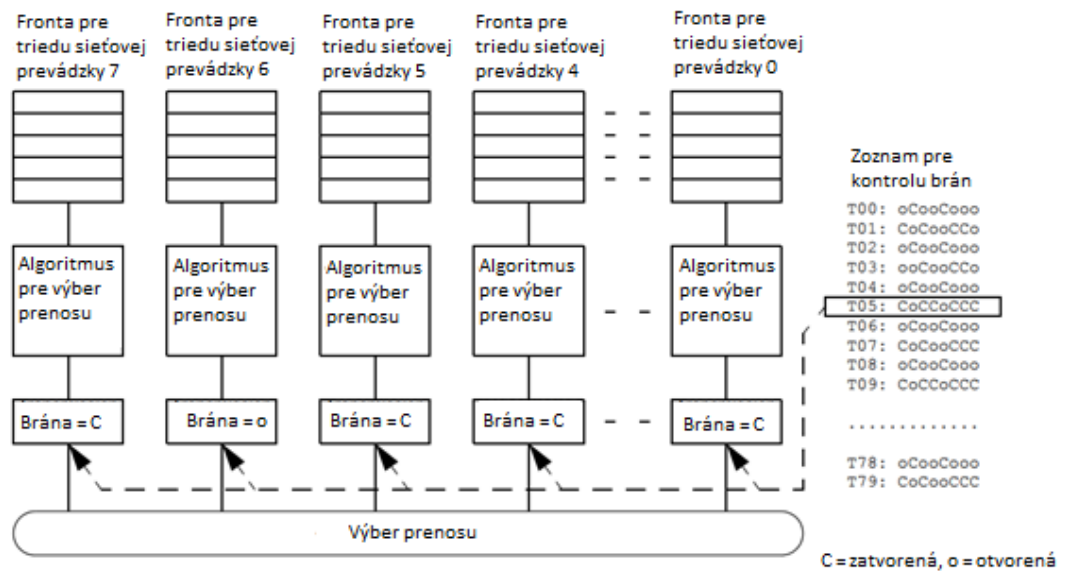
2.1.3 Plánovaná prevádzka a koncové stanice

Mosty by mal podporovať vylepšenia pre pravidelnú leteckú dopravu, zatiaľ čo koncové stanice nie sú spojené s týmito mostami. Plánovaný mechanizmus nesmie byť implementovaný v koncových staniách, to znamená, že koncové stanice sú len príjemcami pravidelnej prevádzky. [7]

Ak je čas nového cyklu pre gating cyklus (AdminCycleTime) ten istý ako čas pre starý gating cyklus (OperCycleTime) a ak je základný čas zvolený pre nový gating cyklus (AdminBaseTime) je deliteľný s OperCycleTime a zároveň odlišný od premennej OperBaseTime, potom sa nový gating cyklus bude presne zrovnávať so starým gating cyklom. Pre tieto prípady bude nový gating cyklus inštalačný a bude sa vykonávať bez problémov s časovaním. Vo všetkých ostatných prípadoch sa bude pred začiatkom prvého nového cyklu posledný starý cyklus zvyčajne skrátiť. Pre prípad veľmi krátkeho posledného cyklu je lepšie predposledný starý cyklus predĺžiť posledným starým cyklom, toto je povolené premennými, ako je CycleTimeExtension. [7]

2.2 Prepojenie vo vnútri 802.1QBV

Na Obr. 2.3 je prepojenie vo vnútri hardwareu popísaného v tomto podštandarde. Zloženie tohto hardwareu je z front, každá trieda sieťovej prevádzky má vlastnú frontu, Transmission Section Algorithm(algoritmus pre selekciu prenosu), rozhoduje ktoré prevádzková trieda má prednosť pred inou, v prípade viac otvorených brán, brány, Gate control list(zoznam kontroly brán), tento zoznam rozhoduje v ktorom čase, ktoré brány budú otvorené a ktoré budú zavreté a Transmission selection, táto časť prepojuje výstup z brán so vstupom do nižšej podvrstvy v rámci linkovacej vrstvy.



Obr. 2.3: Schematické zobrazenie zapojenia pre podštandard QBVčČ.

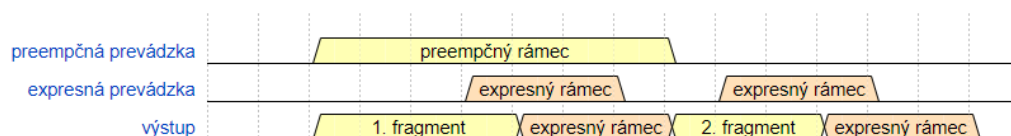
3 802.1QBU

Tento dodatok sa zaoberá preempciou ethernetových rámcov. Ďalej špecifikuje procedúry, spravovanie objektov a rozšírenie protokolov, ktoré sú potrebné pre funkčnú preempciu ethernetových rámcov:

- Definuje triedu služby pre časovo kritické ethernet rámce, ktoré vyžadujú, aby vysielateľ v premostenej lokálnej sieti na prerušenie best effort vysielania, ktoré nie je časovo kritické, pre uvoľnenie prevádzky pre vysielanie časovo kritických rámcov. Po dokončení prenosu časovo kritických rámcov sa obnoví odosielanie prerušeného best effort rámcu. Prerušenie best effort rámcu, ktorý je časovo nekritický, je umožnené viac krát. [8]
- Zabezpečenie vyhľadávania, konfigurácie a kontroly služby preempcií ethernetových rámcov pre mostové porty a koncové stanice. [8]
- Preempcia ethernetových rámcov je zaistená len vtedy, ak v danom prepojení majú obidve zariadenia podporovanú funkciu preempcie ethernetových rámcov. [8]

Cieľom tohto pozmeňujúceho a doplňujúceho dodatku je poskytnúť zníženie latencie prenosu pre plánované časovo kritické rámce v premostenej sieti LAN. Po tomto pozmeňujúcom a doplňujúcom dodatku sa môže spustiť vysielanie rozsiahleho nekritického rámcu pred začiatkom vysielania časovo kritického rámcu. Bez tohto dodatku vzniká nadmerná latencia pre časovo kritické rámce. Veľké množstvo prenosu bez použitia preempcie ethernetových rámcov, bráni schopnostiam aplikácií, ktoré sa používajú pre naplánovaný prenos ethernetových rámcov, na implementáciu riadenia siete v reálnom čase.[8]

Na obr. 3-1 je zobrazená jednoduchá funkcia preempcie ethernetových rámcov. Pokiaľ prebieha prenos ethernet rámcu, ktorý môže byť preemptovaný, a príde na vstup expresný rámec pripravený na vysielanie, funkcie spojené s týmto dodatkom skontrolujú, či počet odoslaných bajtov z ethernet rámcu, pričom musia byť splnené tieto dve podmienky, počet odoslaných bajtov musí byť väčší než je minimálny počet bajtov, stanovený pri konfigurácii a počet bajtov, ktoré ostávajú k prenosu, tiež musí byť väčší ako je minimálny počet bajtov stanovený pri konfigurácii. Po skončení prenosu expresných ethernetových rámcov môže pokračovať prenos preemptovateľných ethernet rámcov



Obr. 3.1: Príklad preempcie rámcov.

3.1 Funkcie 802.1QBU

V nasledujúcich podkapitolách sú popísané spôsoby implementácie, procesy, premenné a funkcie podporované poskytované podštandardom 802.1QBU.

3.1.1 Preemptcia použitá v izolácii

Prerušenie prenosu ethernet rámcu s nižšou prioritou (preemptčný) inými rámcami s vyššou prioritou (expresné) je povolená preemptcia. Prenos prerušeného preemptného rámcu pokračuje po dokončení prenosu expresných rámcov. Pokiaľ je expresný rámec pripravený na vysielanie a pokiaľ je už preemptčný rámec prenášaný, potom oneskorenie expresného rámcu bude v najhoršom prípade doba potrebná na prenos 123 oktetov a často to bude doba potrebná na prenos 64 alebo menej oktetových. Pokiaľ sa preemptcia používa bez plánovania, zníži sa tým chvenie v časovo citlivých rámcach. Všetky triedy sieťovej prevádzky s rovnakou prioritou by mali mať rovnaké označenie. V prípade, že expresné a preemptné priority sú zlúčené do rovnakej triedy sieťovej prevádzky, expresný rámec nemôže zabrániť preemptnému rámcu z tejto fronty. [9]

3.1.2 Plánovanie a preemptcia použitá v kombinácii bez HOLD/RELEASE

Plánovací mechanizmus je popísaný v kapitole 2, ktorá je o podštandarde 802.1QBV, hovorí o vylepšeniach plánovania a "ochrannom pásme". Prenos by mal byť dokončený pred začiatkom prenosu ďalšej triedy. [9]

Pokiaľ sa plánovanie použije s preemptívnymi triedami sieťovej prevádzky, umožní sa preemptnej prevádzke prenášať v ktoromkoľvek čase, potom predbežný prenos rámcu môže začať kedykoľvek pred začiatkom expresného prenosu a môže pokračovať maximálne po dobu potrebnú pre prenos 123 oktetov po skončení prenosu expresnej premávky je pripravený. Toto spôsobí zníženie oneskorenia preemptnej prevádzky, ktoré je spôsobené prenosom expresných rámcov. [9]

3.1.3 Plánovanie a preemptcia použitá v kombinácii s HOLD/RELEASE

Preemptcia ethernet rámcov poskytuje mechanizmus HOLD a RELEASE, tento mechanizmus implementuje explicitné „ochranné pásmo“ okolo chráneného prenosového okna. Požiadavka pre HOLD by mala byť naplánovaná na čas potrebný pre odoslanie 124 bajtov pred začiatkom expresnej prevádzky v chránenom okne, z toho vyplýva, že začiatok vysielania rámcu musí byť maximálne v čase potrebnom na odvysielanie 124 bajtov pred začiatkom chráneného okna a vysielanie sa musí skončiť pred začiatkom chráneného okna. [9]

Použitím kombinácie mechanizmu HOLD a RELEASE, plánovania a preemptcie rámcov spôsobí, že chránené okno je úplne chránené pred rozhraním z predbežnej prevádzky a minimalizuje časový vplyv na predbežnú prevádzku chráneného okna. [9]

3.1.4 Pridelovanie šírky pásma a expresná prevádzka siete

MAC určený v štandarde 802.3 neurčuje, ako dlho môže expresná trieda vykonávať preempciu ethernet rámcu. Je to v rozpore s týmito dvomi podmienkami:

- Ak sa vyskytne nejaká preempčná sieťová prevádzka, existujú požiadavky na oneskorenie vysielania mostu. [9]
- Nie je možné usporiadať expresnú dopravu, aby obsadila celú dostupnú šírku pásma v prípade, že neexistujú neobmedzené fronty. To je opísané teóriou čakania. [9]

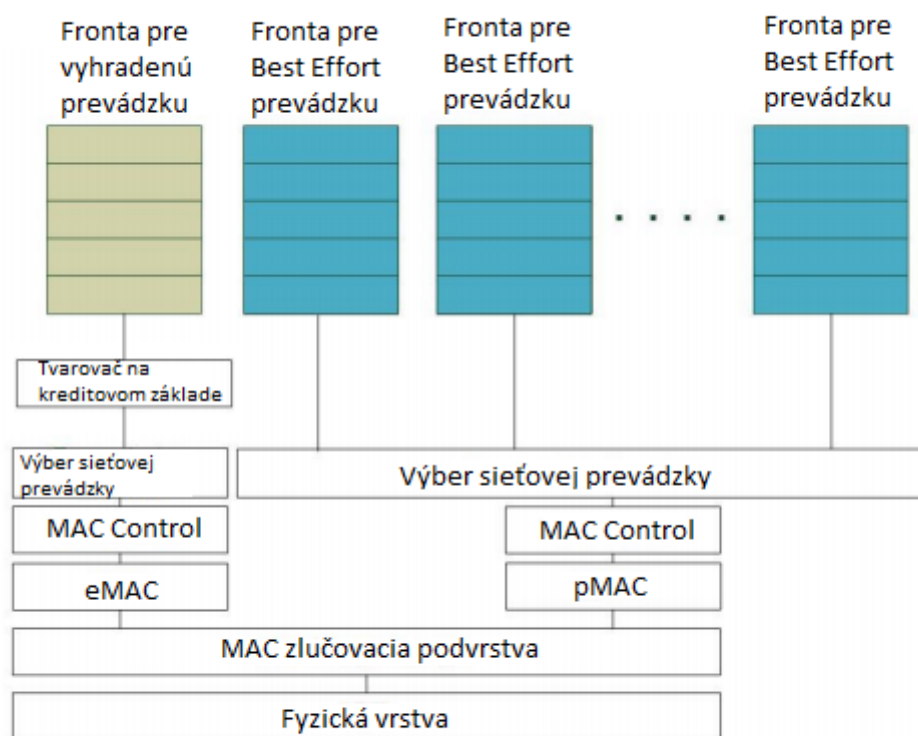
Efektívny mechanizmus preempcie potrebuje rozdelenie šírky pásma menej ako 100% pre expresnú dopravnú triedu podľa časového rozvrhu porovnateľného s oneskoreniami tolerovanými časovo citlivou dopravou. [9]

Preempcia ethernet rámcov môže spôsobiť, že oneskorenie nebude viditeľné na ochranu pred oneskoreným tranzitom. Toto oneskorenie by malo byť oveľa menšie ako požiadavky pre oneskorený prenos. Dalo by sa vyriešiť krátkymi rýchlymi dopravnými oknami a krátkymi medzarami medzi nimi dostatočnými pre čas na maximálne oneskorenie. [9]

Preempcia ethernet rámcu bez plánovania je vhodná na použitie na implementáciu v sieti s očakávaním malého objemu expresnej dopravy. Expresná prevádzka by mala používať tvarovač, ktorý je podobný ako tvarovač založený na kreditoch, s podobnou primeranou väzbou na oneskorenie pri vysielaní. [9]

3.2 Prepojenie v 802.1QBU

Na obr 3-2 je prepojenie vo vnútri hardvéru popísaného v tomto podštandarde. Obsahuje Fronty, jednu pre každú dopravnú triedu, selekciu prenosu, jedna je určená pre expresnú prevádzku a druhá pre preempčnú prevádzku. Táto časť je presne opísaná dodatkom 802.1QBV, dvomi MAC kontrolmi, dvomi MACmi, jedným pre každú prevádzku, MAC Merge Sublayer, ktorý spája obe prevádzky do jednej, a prepojením na fyzickú vrstvu.



Obr. 3.2: Bloková schéma pre 802.1QBU. [12]

4 802.1QBU ŠPECIFIKÁCIA

Pre svoju diplomovú prácu som sa rozhodol implementovať dodatok 802.1QBU a v nasledujúcich podkapitolách sú popísané dôležité funkcie, algoritmy, služby a parametre poskytované a potrebné pre správnu činnosť dizajnu.

4.1 Funkcie pre 802.1QBU

V nasledujúcich podkapitolách sú popísané procesy a funkcie štandardnej 802.1QBU.

4.1.1 Fronty pre ethernet rámce

Tento proces zabezpečuje ukladanie rámcov vo fronte čakajúcich na čas ich odoslania na prenos. Sekvencia rámcov prijatých na rovnakom mostovom porte sa uloží pre unicast rámce s nastaveným VLAN identifikátorom (VID), prioritou a adresou destinácie a zdroja a pre rámce multicast s nastavenou hodnotou VID, prioritou a adresou určenia. Mostový port má jeden alebo viac front, ktoré zodpovedajú odlišnej triede prevádzky. Tabuľka Triedy sieťovej prevádzky mapuje rámce na prevádzku. V tabuľke 4.1 sa odporúča mapovanie počtu implementovaných tried. [10]

Tabuľka 4-1: Doporučené mapovanie priorít pre triedy sieťovej prevádzky. [10]

		Počet dostupných tried sieťovej prevádzky							
		1	2	3	4	5	6	7	8
Priorita	0	0	0	0	0	0	1	1	1
	1	0	0	0	0	0	0	0	0
	2	0	0	0	1	1	2	2	2
	3	0	0	0	1	1	2	3	3
	4	0	1	1	2	2	3	4	4
	5	0	1	1	2	2	3	4	5
	6	0	1	2	3	3	4	5	6
	7	0	1	2	3	4	5	6	7

4.1.2 Výber prenosu

Triedy sieťovej prevádzky podporované portom, vyberú rámev pre prenos. Fronty podporované na porte a stav prenosových brán spojených s týmito frontami odpovedajú algoritmu pre výber prenosu. Rámce sú vyberané pre port a podporovaná hodnota dopravnej triedy je vybraná pre prenos z príslušnej fronty, pokiaľ sú splnené všetky tieto podmienky: [10]

- Rámec vo fronte je pripravený na prenos a určuje ho algoritmus pre výber prenosu. [10]

- Brána pre určovanie prenosu danej fronty musí byť v stave otvorená. [10]
- Pokiaľ sú udalosti na bráne spojené s touto frontou, potom je k dispozícii dostatok času na prenos celého rámcu pred ďalším stavom brány, zatvorená, priradená k tejto fronte [10]
- Pre každú frontu zodpovedajúcu vyššej hodnote dopravnej triedy podporovanej portom, algoritmus pre výber prenosu určuje, že pre prenos v tejto rade nie je k dispozícii žiadny rámec. [10]

Rámec s prioritou n nie je k dispozícii pre prenos, pokiaľ je táto priorita pozastavená na portu s podporou Kontroly prietoku dát založenom na prioritě PFC (Priority-based Flow Control). Keď sa výber prenosu spustí nad agregáciou odkazu (Link Aggregation) a ak fyzický port pozastaví túto prioritu, rámec s touto prioritou nie je k dispozícii. Tabuľka 4.2 zobrazuje Algoritmus pre výber prenosu pre daný port priradený pre triedy, ktoré sú podporované týmto portom. Táto tabuľka je špecifikovaná predávajúčim. [10]

Tabuľka 4-2: Identifikátori algoritmu pre výber prenosu. [10]

Algoritmus pre výber prenosu	Identifikátor
Striktná priorita	0
Tvarovač založený na kreditnom systéme	1
Vylepšený výber prenosu	2
Vyhradené pre budúcu štandardizáciu	3-25
Hodnota algoritmu výberu prenosu špecifického pre dodávateľa pre použitie s DCBX	225
Špecifikácie dodávateľa	Štyri-oktetové celé číslo, kde 3 najvýznamnejšie oktety majú hodnotu OUI a najmenej významný oktet má celočíselnú hodnotu v rozsahu 0-255 pridelený vlastníkom OUI.

4.1.3 Algoritmus pre tvarovač založený na kreditovom systéme

Pokiaľ sú splnené všetky z nasledujúcich podmienok, algoritmus pre tvarovač založený na kreditovom systéme vytvára rámec, vo fronte podporujúcej tento algoritmus, dostupný na prenos:

- Fronta obsahuje aspoň jeden rámec [10]
- Signál transmitAllowed nadobúda hodnotu TRUE [10]

4.1.3 Algoritmus pre tvarovač založený na kreditovom systéme používa nasledujúce externé parametre, ktoré sú spojené s frontou:

- portTransmitRate, operácia v MAC určuje hodnotu prenosovej rýchlosti, je to v bitoch za sekundu. [10]

- *idleSlope*, je miera zmeny kreditu, stúpanie hodnoty kreditu stúpa, zatiaľ čo prenos je FALSE. Hodnota *idleSlope* nikdy nemôže byť vyššia ako hodnota *portTransmitRate*. Hodnota parametra *operIdleSlope* určuje hodnotu *idleSlope* pre aktuálnu frontu. [10]

Algoritmus pre tvarovač založený na kreditovom systéme používa nasledujúce interné parametre, ktoré sú spojené s frontom:

- Parameter *transmit*, počas vysielania rámcov z fronty, tento parameter má hodnotu TRUE, po dokončení prenosu rámca sa hodnota parametru zmení na FALSE. V kombinácii s týmto algoritmom a predplatením rámca je táto hodnota TRUE, zatiaľ čo MAC prenesie rámček. Po oneskorení alebo prerušení prenosu rámca sa vysielanie nastaví na hodnotu FALSE, až kým začne alebo nepretrží prenos rámca. Réžia prevodovky, ktorá je vytvorená rámcom prefakturovania. [10]
 - Parameter *credit*, tento parameter je k dispozícii vo fronte. Pokiaľ neexistuje žiadny rámec a parameter *transmit* je nastavený na hodnotu FALSE a hodnota parametru *credit* je väčšia ako nula a vo frontoch nie je preemptovateľný rámec s prebiehajúcim prenosom, ale prenos bol pozastavený, potom hodnota parametru *credit* klesne na nulu. [10]
 - Parameter *endSlope*, tento parameter je miera zmeny parametru *credit* po znížení hodnoty parametru *credit*. Tento parameter je definovaný v rovnici 4.1. [10]
- $$\text{endSlope} = \text{idleSlope} - \text{portTransmitRate} \quad (4.1)$$
- Parameter *transmitAllowed*, tento parameter má hodnotu TRUE, keď je parameter *credit* vyšší alebo rovný nule, inak je tento parameter nastavený na hodnotu FALSE. [10]

4.1.4 Vylepšený algoritmus pre výber prenosu

Po aktivácii vylepšeného algoritmu pre výber prenosu ETS (Enhanced Transmission Selection) pre zvolenú triedu sieťovej prevádzky, sa prenos zvolí priradením šírky pásma tejto triede sieťovej prevádzky. ETS umožňuje plánovať súvisiace kontrolné operácie. [10]

Tento algoritmus určí rámec vo fronte ako pripravený na prenos, ak sú splnené všetky nasledujúce podmienky:

- Vo fronte je jeden alebo viac rámcov. [10]
- Rámec by sa mal vysielat' z fronty, ak to ETS určí. [10]
- Fronty so striktným prioritným algoritmom alebo algoritmom založený na kreditovom systéme nemajú pripravené pre prenos žiadne rámce. [10]

4.1.5 Preempcia ethernet rámcov

Nasledujúce podmienky musia byť aplikované pre všetky porty ktoré podporujú preempciu ethernet rámcov:

- Každá trieda sieťovej prevádzky, ktorá je podporovaná portom, má priradenú hodnotu stavu preempčného rámca. Možné hodnoty stavu preempčného rámca sú buď preempťovaný alebo preempťovateľný. [10]
- Dve inštancie MAC podporujú port (preempťovateľná inštancia MAC služby a inštancia služby preempťovaná MAC) [10]

4.2 MAC service

Pre preempciu ethernet rámca je potrebné aby MAC mal dve MAC servisné rozhrania:

- Preempčné MAC (pMAC) servisné rozhranie [9]
- Expresné MAC (eMAC) servisné rozhranie [9]

4.3 Parametre potrebné pre funkciu preempciu ethernet rámcov

V nasledujúcich podkapitolách sú popísané parametre potrebné na riadenie procesu preempcie ethernet rámcu.

4.3.1 framePreemptionStatusTable štruktúry a dátové typy

“framePreemptionStatusTable pozostáva z 8 hodnôt framePreemptionAdminStatus, jeden na jednu prioritu. ” [9]

FramePreemptionAdminStatus je hodnota pre každú prioritnú frontu. Ak je táto hodnota expresná, táto priorita využíva expresnú službu. Ak je táto hodnota preempčná, táto priorita využíva preempčné služby. [9]

4.3.2 Objekt s holdAdvance

Objekt holdAdvance je celočíselná hodnota, ktorá predstavuje maximálny počet nanosekúnd, ktorý môže uplynúť medzi požiadavkou HOLD na MAC a MAC pozastavením prenosu akéhokoľvek rámca. [9]

4.3.3 Objekt s releaseAdvance

Uvoľňovací objekt releaseAdvance je celočíselná hodnota, ktorá predstavuje maximálny počet nanosekúnd, ktorý môže uplynúť medzi požiadavkou REALEASE na MAC a MAC pokračovaním prenosu akéhokoľvek rámca. [9]

4.3.4 Objekt s preemptionActive

Tento objekt je boolean, čo indikuje podporu predbežných rámcov s MAC. Je nastavená na hodnotu TRUE, ak ju podporuje MAC. [9]

4.3.5 Objekt s holdRequest

“Objekt holdRequest obsahuje vyčíslenú celočíselnú hodnotu, pričom pozdržanie (1) indikuje, že bola vykonaná operácia brány Set-And-HoldMAC a uvoľnenie (2) indikovalo, že bola vykonaná operácia brány Set-And-ReleaseMAC. Tento objekt existuje na každom porte. " [9]

5 802.3BR

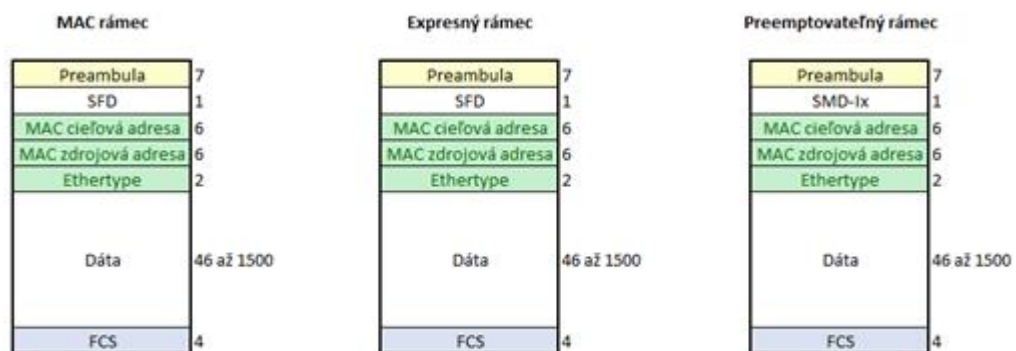
Pre kódovanie preempčnej prevádzky sa nepoužíva štandardné kódovanie, ako pri klasickom ethernet, ale kódovanie popísané v štandarde 802.3BR.

5.1 Formát rámcu podľa 802.3BR

Expresné rámce majú formát ako klasické ethernet rámce, sedem bajtov preambula 0x55, jeden bajt SFD 0xD5, šesť bajtov MAC cieľová adresa, šesť bajtov MAC zdrojová adresa, dva bajty ethertype, dátové pole s veľkosťou 46 až 1500 bajtov a na konci FCS s iniciálnou hodnotou 0xFFFFFFFF.[11]

Preempčné rámce, bez fragmentácie sa trochu odlišujú, namiesto SFD sa používa SMD-Ix(start mframe delimiter, initial fragment). [11]

Porovnanie formátu expresných a preempčných rámcov s formátom klasického ethernet rámcu je zobrazené na Obr.5.1. Napravo od grafického modelu rámcu sú napísané počty bajtov, z ktorých sa skladá daná časť rámcu. [11]



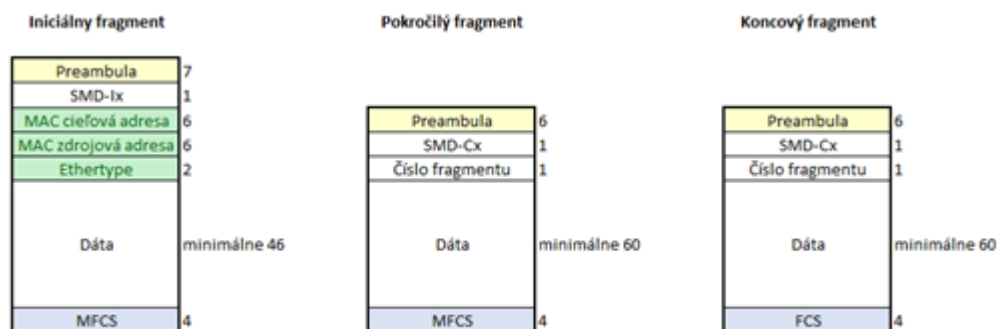
Obr. 5.1: Porovnanie formátov klasického MAC rámcu, Expresného rámcu a Preemptovateľného rámcu. [11]

Pri fragmentácii preempčného rámcu, vyvolanej preempciou, fragmentovať sa môžu len rámce dlhšie ako 128 bajtov bez preamble a SMD a to z dôvodu, že každý fragment musí byť minimálne tak dlhý ako je minimálna dĺžka ethernet rámcu bez preamble a SFD, čo je 64 bajtov, môžu vzniknúť až tri typy fragmentov preempčného rámcu: [11]

- Iniciálny fragment, tento fragment je identický s nefragmentovaným preempčným rámcom až na to, že koncová štvorbajtová sekvencia MFCS (mframe check sequence) sa na rozdiel od FCS má iniciálnu hodnotu 0xFFFF0000. [11]
- Pokročilé fragmenty, tieto fragmenty sa nachádzajú medzi prvým (iniciálnym) fragmentom a posledným fragmentom daného rámcu, skladajú sa zo šiestich bajtov preamble, namiesto SMD-Ix majú SMD-Cx(start mframe delimiter continue fragment), číslo fragmentu. dátová časť s minimálnou dĺžkou 60 bajtov a MFCS. [11]

- Posledný fragment, tento fragment začína rovnako ako pokročilé fragmenty, dátová časť, tiež musí byť minimálne 60 bajtov dlhá, ale na konci tohto rámcu je FCS celého rámcu. [11]

Na Obr.5.2 sú zobrazené všetky tri možnosti fragmentov rámcu.



Obr. 5.2: Formáty fragmentov preemptovateľného rámcu. [11]

5.2 Kódovanie SMD a čísla fragmentu

SMD-Ix môže nadobúdať štyri hodnoty, ktoré sa neustále cyklia. Po odoslaní celého rámcu sa táto hodnota posunie. [11]

SMD-Cx a číslo fragmentu tiež môžu nadobúdať štyri hodnoty ktoré sa tiež neustále cyklia, ale po odoslaní fragmentu. [11]

Všetky tieto hodnoty sú zapísané v tabuľke 5.1.

Tabuľka 5-1: Hodnty pre SMD polia a polia s číslom fragmentu. [11]

Číslo	SMD-Ix	SMD-Cx	Číslo fragmentu
0	0xE6	0x61	0xE6
1	0x4C	0x52	0x4C
2	0x7F	0x9E	0x7F
3	0xB3	0xAD	0xB3

6 TEORETICKÝ VÝPOČET A POROVNANIE PRIEPUSTNOSTI 802.QBU S INÝMI ŠTANDARDMI

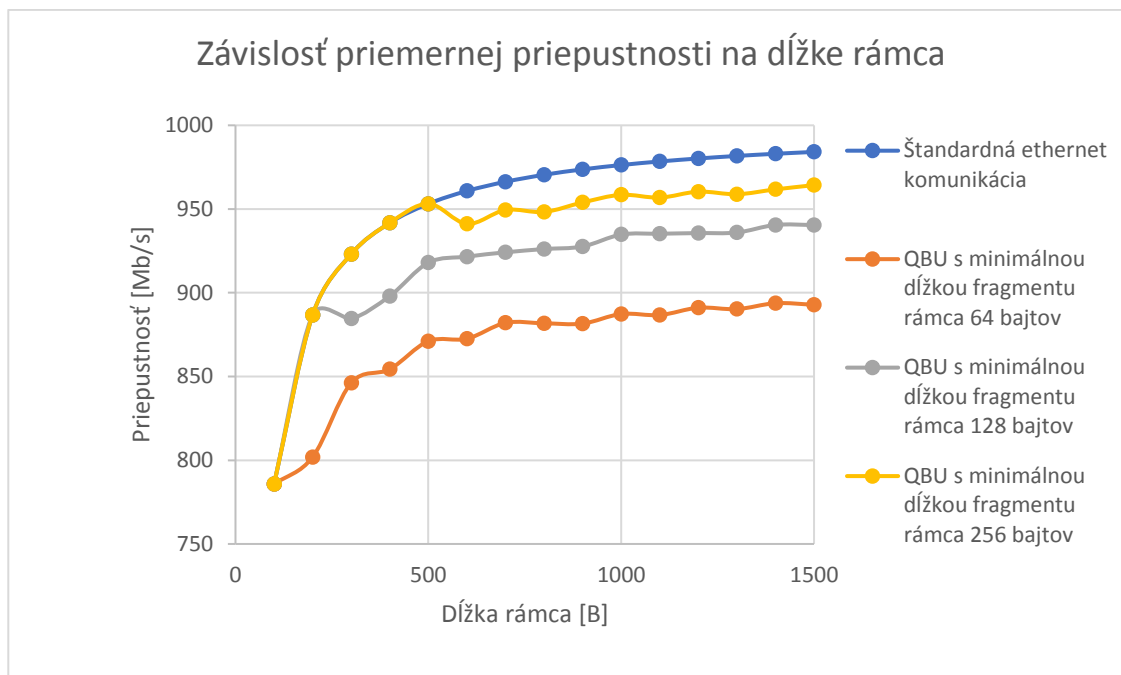
V tejto kapitole sú spočítané teoretické hodnoty priepustnosti navrhovaného štandardu 802.1QBU a iných vybraných štandardov, pre porovnanie s týmto štandardu, pre porovnanie priepustnosti. Ďalej tu je spočítaná maximálna latencia tohto štandardu, pri rôznych implementáciách. Priepustnosť bola spočítaná pomocou vzorca 1.1, spomenutého v prvej kapitole. Ako pravdepodobnosť odoslania chybného rámcu sa zvolila nulová hodnota, keďže tieto štandardy podporujú maximálnu dĺžku rámcu 1500 bajtov a doporučená dĺžka rámcu sa v dnešnej dobe pohybuje okolo 8000 bajtov, takže pravdepodobnosť odoslania chybného rámcu je strašne nízka a je ju možné zanedbať vo výpočtoch.

6.1 Porovnanie priemernej sieťovej prevádzky cez 802.1QBU a cez štandardný ethernet

Porovnanie 802.1QBU so štandardným ethernetom prebehlo, spočítaním priepustnosti možných implementácií (fragment s minimálnym počtom odoslaných dát 64, 128 a 256 bajtov), výsledná priepustnosť vznikla priemerom dvoch hodnôt, najlepšieho prípadu (maximálne jedna preempcia, jedným 64 bajtovým expresným rámcom, počas vysielania dlhého rámcu) a najhoršieho prípadu (maximálny počet preempcií, jedným 64 bajtovým expresným rámcom, počas vysielania dlhého rámcu). Veľkosť dlhého preempčného rámcu sa pohybovala od 100 do 1500 bajtov, s nárastom hodnoty po 100 bajtoch.

Pre výpočet priepustnosti štandardného ethernetu sa počítalo z celkového počtu bajtov odoslaných štandardom 802.1QBU a počtu nadmerných bajtov pre daný počet rámcov. Výsledná priepustnosť sa tiež spočítala spriemerovaním najlepšieho a najhoršieho prípadu.

Do výpočtov boli ďalej zahrnuté tieto čísla: 8 informačných bitov na jeden znak, rýchlosť prenosu údajov bola stanovená na 125 M znakov za sekundu(1 Gb/s) a čas medzi rámcami bol zvolený na 96 ns (12 znakov medzi jednotlivými rámcami).



Obr. 6.1: Závislosť priemernej priepustnosti na dĺžke rámca, pre štandardný ethernet a varianty podštandardu s rôznou minimálnou dĺžkou fragmentu.

Na obr.6.1 je zobrazené grafické porovnanie, závislosti priepustnosti na veľkosti dlhého rámca, možných implementácií štandardu 802.1QBU, s rôznou minimálnou dĺžkou, a štandardného ethernetu. U štandardu 802.1QBU veľkosť priepustnosti rastie s minimálnou dĺžkou fragmentu, čo je možné vysvetliť menšou fragmentáciou rámca, čiže menším počtom pridaných znakov v rámci jedného rámca.

6.2 Porovnanie uvažovanej sieťovej prevádzky cez ethernet, 802.1QBU, 802.1QBV a ich kombinácii

Ako uvažovaná sieťová prevádzka sa uvažovala prevádzka, kde na jeden preempčný rámec (dlhý rámec s nižšou prioritou) pripadá jeden expresný (krátky rámec dlhý 64 bajtov). Výpočty priepustnosti sa počítali obdobne ako v predchádzajúcej podkapitole.

Pre štandardný ethernet sa preto počítala priepustnosť z dvoch rámcov vysielaných po sebe.

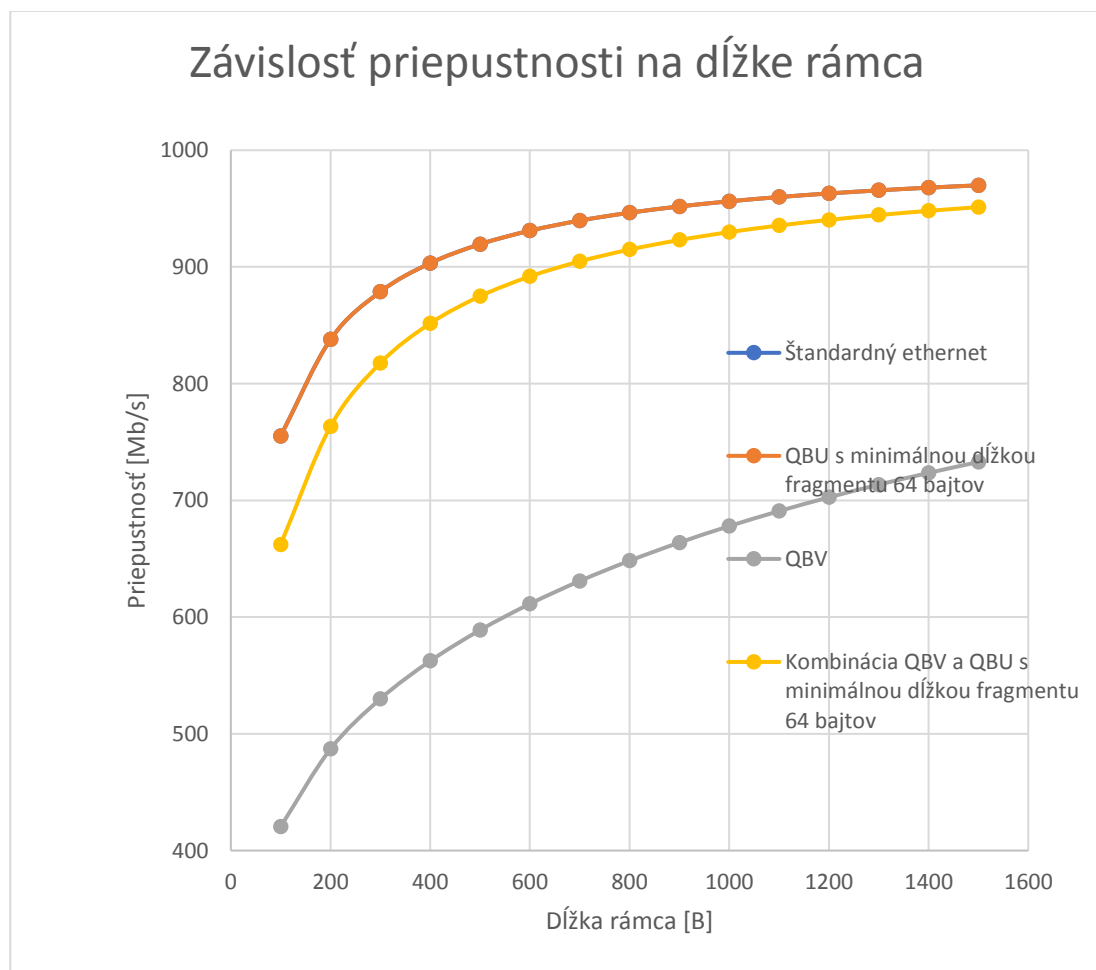
Pre štandard 802.1QBU sa preto uvažovala iba jedna preempcia na preempčný rámec.

Pre štandard 802.1QBV sa ako výsledná hodnota priepustnosti počítala ako priemer hodnoty priepustnosti s maximálnou dobou čakania, to jest čas potrebný na odoslanie rámca s maximálnou dĺžkou (1526 bajtov), a s nulovou dobou čakania, dokonalo naplánovaná doba odosielania rámcov.

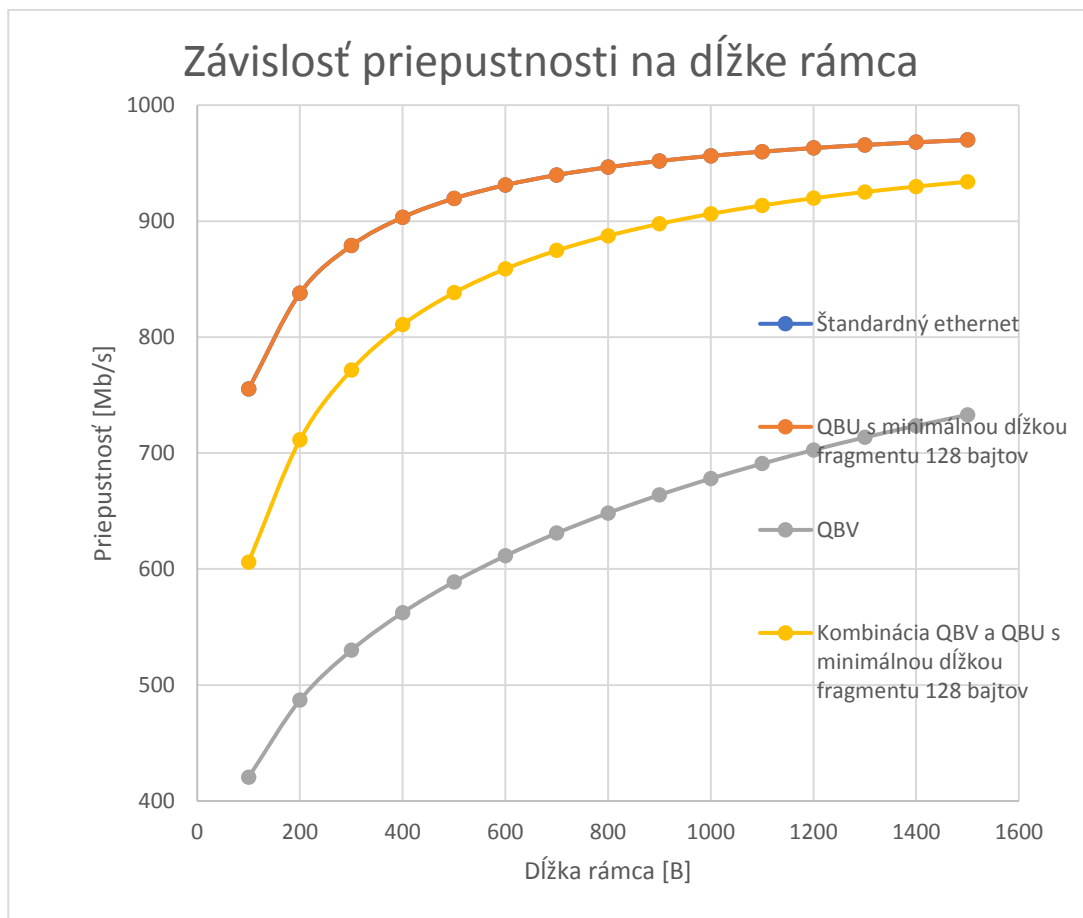
Pri kombinácii štandardov 802.1QBU a 802.1QBV sa ako výsledná hodnota priepustnosti počítala ako priemer hodnoty priepustnosti s maximálnou dobou čakania, to

jest čas potrebný na odoslanie fragmentu rámcu s maximálnou dĺžkou (64 až 128 bajtov, pre minimálnu dĺžku fragmentu 64 bajtov, 128 až 256 bajtov, pre minimálnu dĺžku fragmentu 128 bajtov a 256 až 512 bajtov, pre minimálnu dĺžku fragmentu 256 bajtov), a s nulovou dobou čakania, dokonalo naplánovaná doba odosielať rámcov.

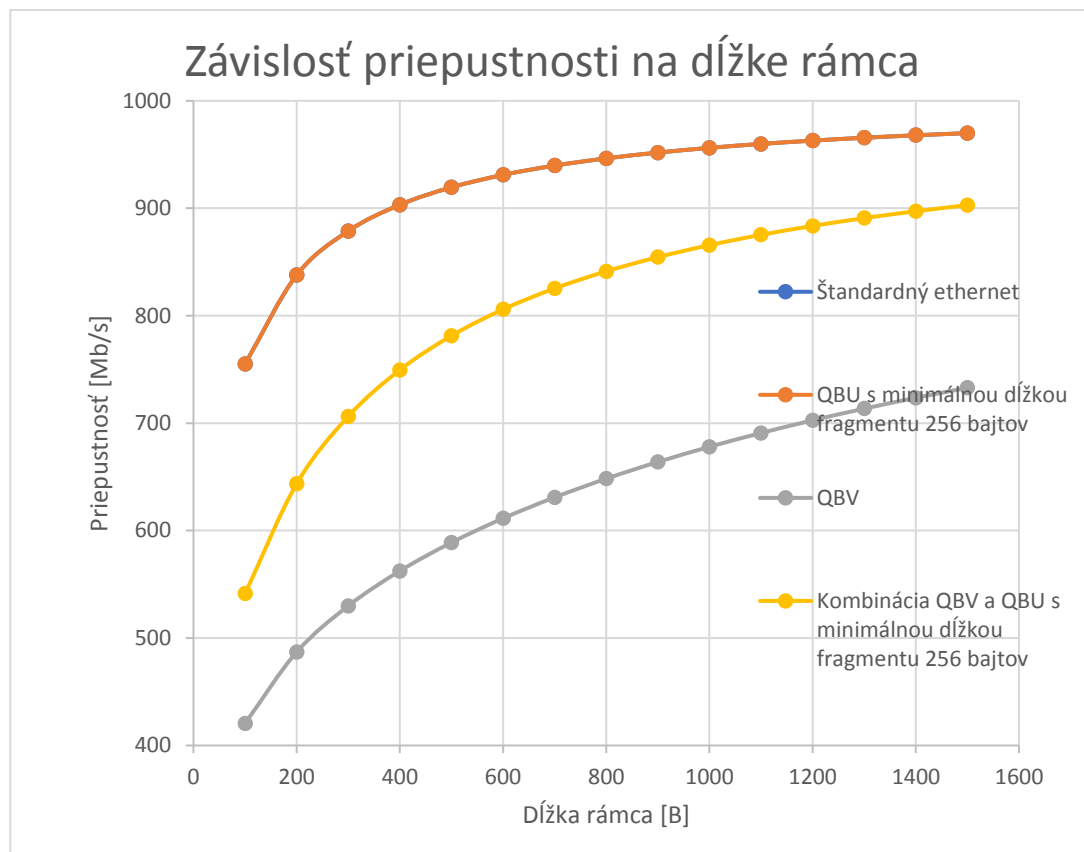
Na obr.6.2 zobrazené grafické porovnanie, závislosti priepustnosti na veľkosti dlhého rámcu, s minimálnou dĺžkou fragmentu 64 bajtov. Na obr.6.3 zobrazené grafické porovnanie, závislosti priepustnosti na veľkosti dlhého rámcu, s minimálnou dĺžkou fragmentu 128 bajtov. Na obr.6.4 zobrazené grafické porovnanie, závislosti priepustnosti na veľkosti dlhého rámcu, s minimálnou dĺžkou fragmentu 256 bajtov. Na všetkých grafoch je možné vidieť, že priepustnosť štandardného ethernetu a 802.1QBV sa nemení v závislosti na minimálnej dĺžke fragmentu, keďže u nich neprebíha fragmentácia rámcu. Štandard 802.1QBU sa priepustnosťou najviac približuje štandardnému ethernetu. Priepustnosť kombinácie 802.1QBU a 802.1QBV sa pohybuje medzi priepustnosťami štandardu 802.1QBV a štandardného ethernetu, priepustnosť klesá s rastúcou minimálnou dĺžkou fragmentu.



Obr. 6.2: Porovnanie závislostí priepustností pre štandardný ethernet QBV, QBU s minimálnou dĺžkou fragmentu 64 bajtov a kombinácie QBV s QBU s minimálnou dĺžkou fragment 64 bajtov.



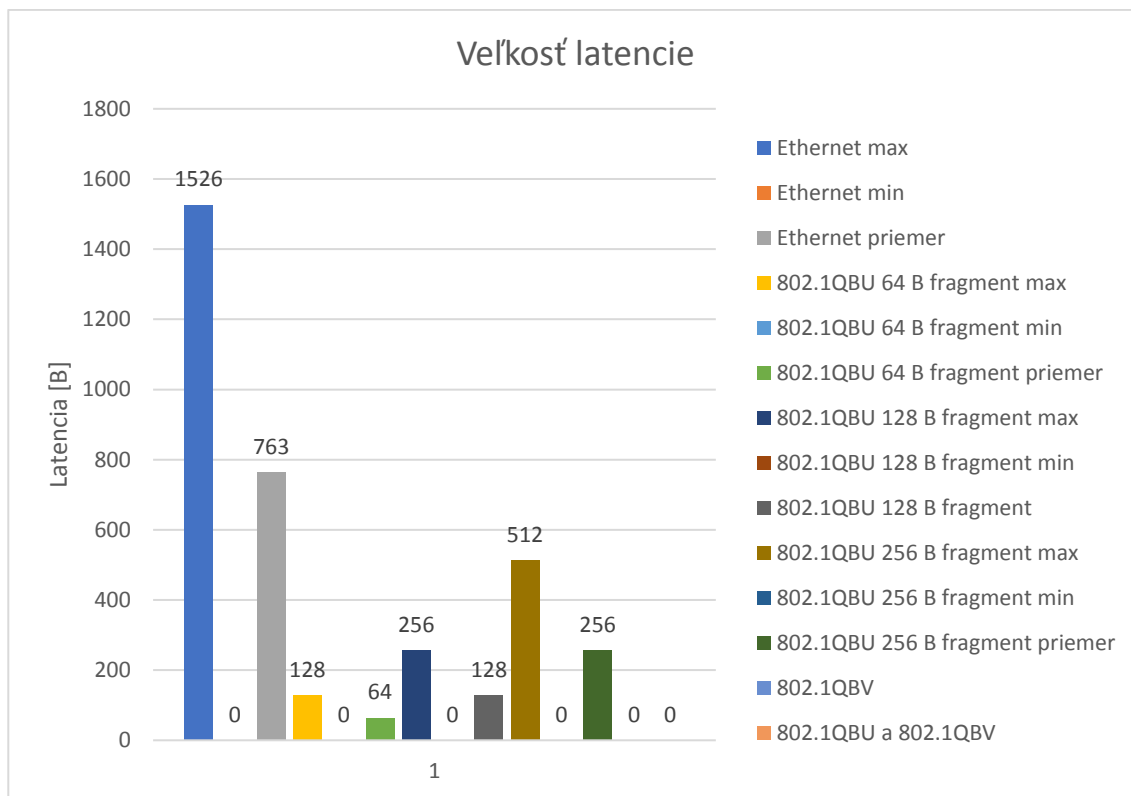
Obr. 6.3: Porovnanie závislostí priepustností pre štandardný ethernet QBV, QBU s minimálnou dĺžkou fragmentu 128 bajtov a kombinácie QBV s QBU s minimálnou dĺžkou fragment 128 bajtov.



Obr. 6.4: Porovnanie závislostí priepustností pre štandardný ethernet QBU, QBU s minimálnou dĺžkou fragmentu 256 bajtov a kombinácie QBV s QBU s minimálnou dĺžkou fragment 256 bajtov.

6.3 Porovnanie latencie

Na obr.6.5 je graf kde sú zobrazená latencia expresných rámcov (časovo kritické rámce) porovnávaných štandardov sieťovej komunikácie. Sieťové štandardy 802.1QBV a kombinácia štandardov 802.1QBU a 802.1QBV majú nulovú latenciu, pretože čas vysielania expresných rámcov je naplánovaný a počas ochranného pásma nezačína žiadna nová komunikácia. Pri komunikácii cez štandardný ethernet môže latencia časovo kritických rámcov dosahovať až čas potrebný pre odoslanie rámcu s maximálnou dĺžkou, preto je štandardný ethernet nevhodný pre aplikácie s časovo kritickou triedou sieťovej prevádzky. Latencia u štandardu 802.1QBU môže dosahovať od nulovej doby po čas potrebný na odoslanie rámcu s dvojnásobnou dĺžkou ako je minimálna dĺžka fragmentu.



Obr. 6.5: Porovnanie veľkosti latencie pre expresné rámce.

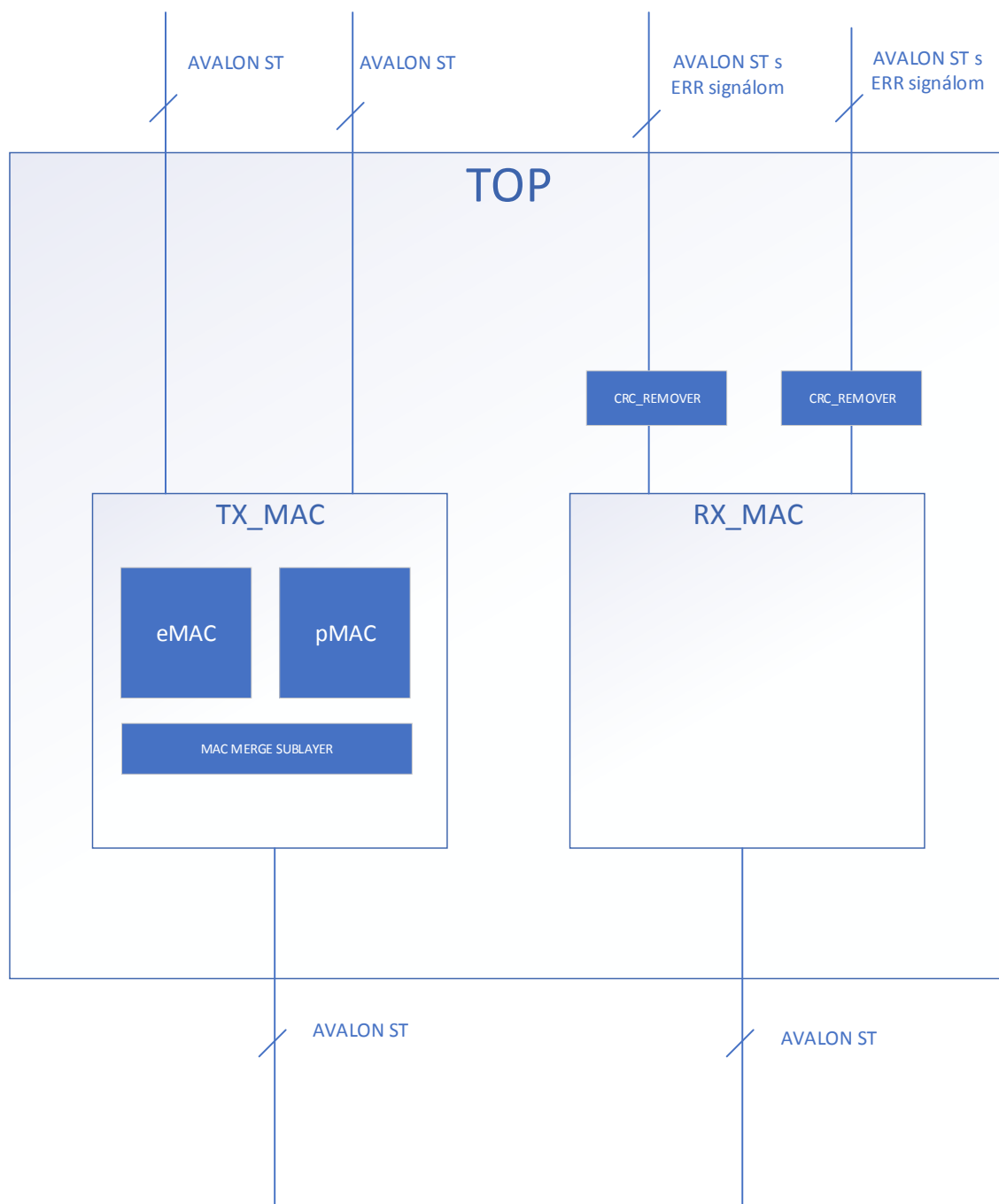
7 IMPLEMENTÁCIA PODŠTANDARDU 802.1QBU

Pre implementáciu bola zvolená preempcia v izoláciu s dvomi triedami sieťovej prevádzky zo strany hosta, keďže podľa štandardu sa triedy sieťovej prevádzky podľa priority a konfigurácie priradujú do dvoch tried sieťovej prevádzky, expresnej a preempčnej, na vyššej podvrstve. Preempcia v izolácii bola zvolená z dôvodu jednoduchšej implementácie a nie je pri nej potrebné implementovať iný rozširujúci podštandard zo skupiny 802.1Q. Pre posielanie paketov medzi IP v FPGA sa používajú rozhrania ako Avalon ST a AXI. Pre vyššiu flexibilitu a pomocné signály, ktoré je možné zahrnúť do rozhrania Avalon ST sa ako rozhranie na vstupoch a výstupoch zvolilo ako prenosové rozhranie Avalon ST. Rozhranie Avalon ST má štandardne štyri signály (clk, data, valid a ready), pomocou signálu valid source (vysielač) oznamuje pre sink, že sú na vstupe platné dáta, sink (príjemca) pomocou signálu ready dáva informáciu pre source, že je pripravený na príjem dát. Pre tento dizajn sa použili rozširujúce signály start of packet a end of packet, to z dôvodu že pri informácii, kedy dátový paket začína a kedy končí, je možné efektívnejšie navrhnuť dizajn. [13]

TX strana plní nasledujúce funkcie, prichádzajúce dátové pakety zabaľuje do ethernet rámcov, počíta CRC ktoré sa umiestni na koniec rámca do FCS poľa, určuje prioritu tried sieťovej prevádzky v prípade rovnakého času príchodu dátových paketov, preempciu rámcov pri vysielaní preempčnej triedy sieťovej prevádzky a príchode dátového paketu na expresnej triede, ale až po splnení týchto dvoch podmienok, odoslaný počet bajtov je väčší ako dĺžka fragmentu s minimálnou dĺžkou, čo je 64 bajtov (60 bajtov ak sa nepočíta dĺžky FCS) a počet zvyšných bajtov na odoslanie musí byť väčší ako je minimálna dĺžka fragmentu. Funkcia, ktorá bola implementovaná nadštandardne, je rozdelenie rámcu s dĺžkou presahujúcou maximálnu dĺžku rámca, čo je 1520 bajtov, ho rozdeliť na viac rámcov. V prípade rámcov, ktoré sú kratšie ako minimálna dĺžka rámca, čo je 64 bajtov, sa za ich dátovú časť pripája takzvaný padding, ako dáta sa dosadia samé nuly, kým sa nedosiahne minimálna požadovaná hodnota. Na Obr. 7.1 je v TX MAC znázornené bloky eMAC, pMAC a MAC merge sublayer, ktoré však slúžia pre logické zobrazenie funkcií, obsiahnutých v TX MAC. eMAC (expresný MAC) sa vyznačuje rovnakými funkciami ako štandardný ethernetový MAC, čo je zabaľovanie dátových paketov do rámcov a počítanie CRC pre FCS pole a dodržiavanie vzdialenosti medzi dvomi rámcami. pMAC (preempčný MAC) má tiež vlastnosti ako štandardný MAC, plus funkciu prerušenia vysielania. MAC merge sublayer slúži ako multiplexor medzi výstupmi z eMAC a pMAC, pričom eMAC má vždy vyššiu prioritu.

RX strana naopak plní tieto funkcie, u prichádzajúcich rámcov kontroluje ich formát, či sedí preambula, SFD, SMD a číslo fragmentu, podľa toho či prišlo za preambulou SFD alebo SMD, triedi rámce buď do expresnej triedy, ak prišlo SFD, alebo do preempčnej triedy, ak prišlo SMD. Kontroluje či prichádzajúce rámce majú správnu dĺžku, v prípade nesprávnej dĺžky rámca RX strana propaguje chybu vyššej vrstvy, pri dlhšom rámci prestane preposielať dáta z rámca a nastaví chybu. Ďalej kontroluje, či na konci každého rámca alebo fragmentu je pridané pole so správnym FCS, v prípade nesprávneho FCS oznámi vyššej vrstve, že daný rámec nie je správny. RX strana tiež za pomoci FCS vie určiť, či prišiel celý rámec, jeho fragment alebo posledný fragment rámca. Ak neprišiel celý rámec alebo konečný fragment, nepošle vyššej vrstve informáciu o konci rámca

v podobe signálu end of packet.



Obr. 7.1: Top modul implementovaného dizajnu, s prepojením vnútorných komponentov.

7.1 RX MAC

RX MAC príma rámce a kontroluje, či počas prenosu nedošlo k chybe prenosu pomocou polynómu pre výpočet CRC, pri prijatí správneho rámcu vždy ako posledná hodnota vyjde konštanta, čo pri použitom algoritme v tomto kóde je 32 bitové číslo

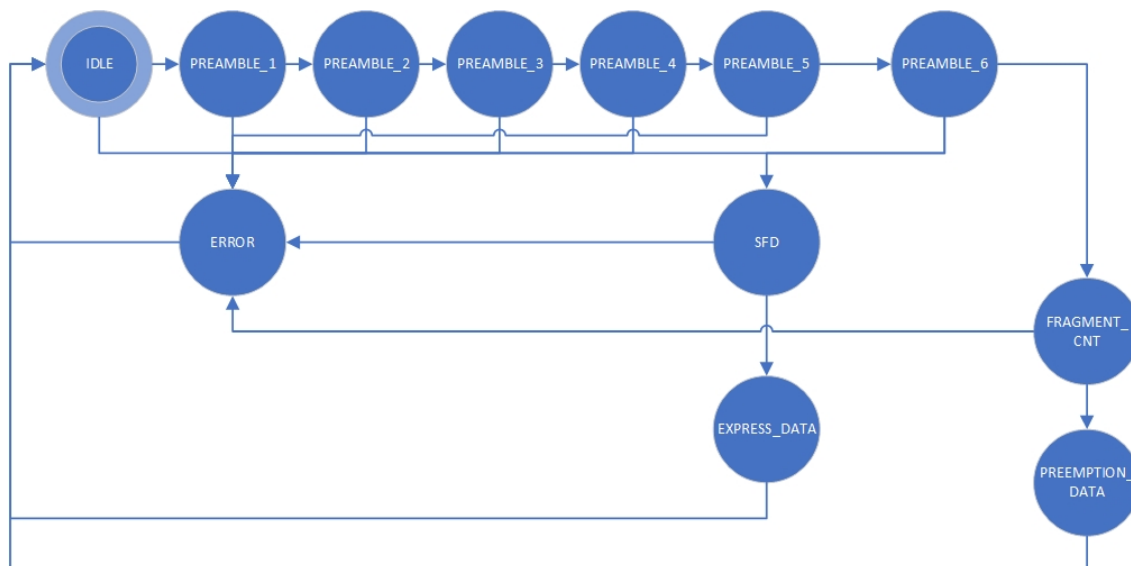
s hexadecimálnou hodnotou 0x DEBB20E3. Začiatok rámcu oznamuje pomocný signál pre rozhranie AVALON Streaming sop (start of packet) po obdržaní tohto signálu a úrovne 1 v potvrdzovacom signáli valid sa začne kontrola bitov v preamble, SFD alebo SMD a poradia fragmentu (pokiaľ sa nejedná o prvý fragment). V prípade neúspešnej kontroly týchto polí, nastane stav error, kedy sa dáta začnú zahadzovať až do príchodu pomocného signálu pre AVALON ST eop (end of packet). V opačnom prípade sa podľa SFD alebo SMD začnú prenášať vstupné dáta buď na expresnú triedu sieťovej prevádzky, ak príde SFD, alebo na preempčnú triedu sieťovej prevádzky, ak príde SMD. Po príchode aktívnej úrovne signálu eop sa skontroluje konštanta pre CRC, v prípade odlišnosti od konštanty sa na danú triedu sieťovej prevádzky vygeneruje signál podporovaný rozhraním AVALON ST error.

Tabuľka 7-1: Tabuľka prechodov medzi stavmi pre automat v TX MAC.

Stav	nasledujúci stav	podmienka
IDLE	PREAMBLE_1	sop=1 and data=0x55
PREAMBLE_1	PREAMBLE_2	data=0x55
	ERR_DATA	data=/0x55
PREAMBLE_2	PREAMBLE_3	data=0x55
	ERR_DATA	data=/0x55
PREAMBLE_3	PREAMBLE_4	data=0x55
	ERR_DATA	data=/0x55
PREAMBLE_4	PREAMBLE_5	data=0x55
	ERR_DATA	data=/0x55
PREAMBLE_5	PREAMBLE_5	data=0x55
	ERR_DATA	data=/0x55
PREAMBLE_6	PREAMBLE_6	data=0x55
	ERR_DATA	data=/0x55 and data=/SMDc
	FRG_CNT	data=SMDc
SFD	EXPRESS_DATA	data=SFD
	ERR_DATA	data=/SFD and data=/SMDi
	PREEMPCTION_DATA	data=SMDi
EXPRESS_DATA	IDLE	eop=1
	ERR_DATA	e_size > MAX_FRAME_DATA_BYTES + CRC32_BYTES
PREEMPCTION_DATA	IDLE	eop=1
	ERR_DATA	p_size > MAX_FRAME_DATA_BYTES + CRC32_BYTES
FRG_CNT	PREEMPCTION_DATA	data=next_frg
	ERR_DATA	data=/next_frg
ERR_DATA	IDLE	eop=0

Tabuľka 7-2: Tabuľka výstupov pre stavový automat v RX MAC.

Stav	Preempčná prevádzka				Expresná prevádzka			
	data	sop	eop	valid	data	sop	eop	valid
IDLE	0x00	0	0	0	0x00	0	0	0
PREAMBLE_1	0x00	0	0	0	0x00	0	0	0
PREAMBLE_2	0x00	0	0	0	0x00	0	0	0
PREAMBLE_3	0x00	0	0	0	0x00	0	0	0
PREAMBLE_4	0x00	0	0	0	0x00	0	0	0
PREAMBLE_5	0x00	0	0	0	0x00	0	0	0
PREAMBLE_6	0x00	0	0	0	0x00	0	0	0
EXPRESS_DATA	0x00	0	0	0	data_i	sop_i	eop_i	valid_i
PREEMPTION_DATA	data_i	sop_i	eop_i	valid_i	0x00	0	0	0
FRG_CNT	0x00	0	0	0	0x00	0	0	0
ERR_DATA	0x00	0	0	0	0x00	0	0	0



Obr. 7.2: Diagram stavového automatu v RX MAC.

Prechody medzi stavmi sú vykreslené na obr.7.2. V tabuľke 7.1 sú vypísané výstupy z RX MACu pri danom stave. V tabuľke 7.2 sú popísané prechody medzi stavmi a podmienky, ktoré musia byť dodržané.

7.2 TX MAC

TX MAC vysiela rámce a pridáva na každého rámca začiatok polia s preambulami, SFD, SMD a poradie fragmentu v prípade fragmentácie vysielaného rámca. Na koniec každého rámca pridá štyri kontrolné bajty, tie sú vypočítané z celého rámca alebo fragmentu. TX MAC uprednostňuje dáta z expresnej triedy sieťovej prevádzky, dokáže vykonať preempciu (prerušenie vysielania) preempčnej triedy

sieťovej prevádzky a začať vysielanie dát z expresnej triedy sieťovej prevádzky, ale musia byť dodržané dve podmienky: počet odvysielaných bajtov musí byť väčší ako je minimálna dĺžka fragmentu, čo je v tomto prípade 64 bajtov, a počet zvyšných bajtov na odoslanie musí byť tiež väčší ako je minimálna dĺžka fragmentu.

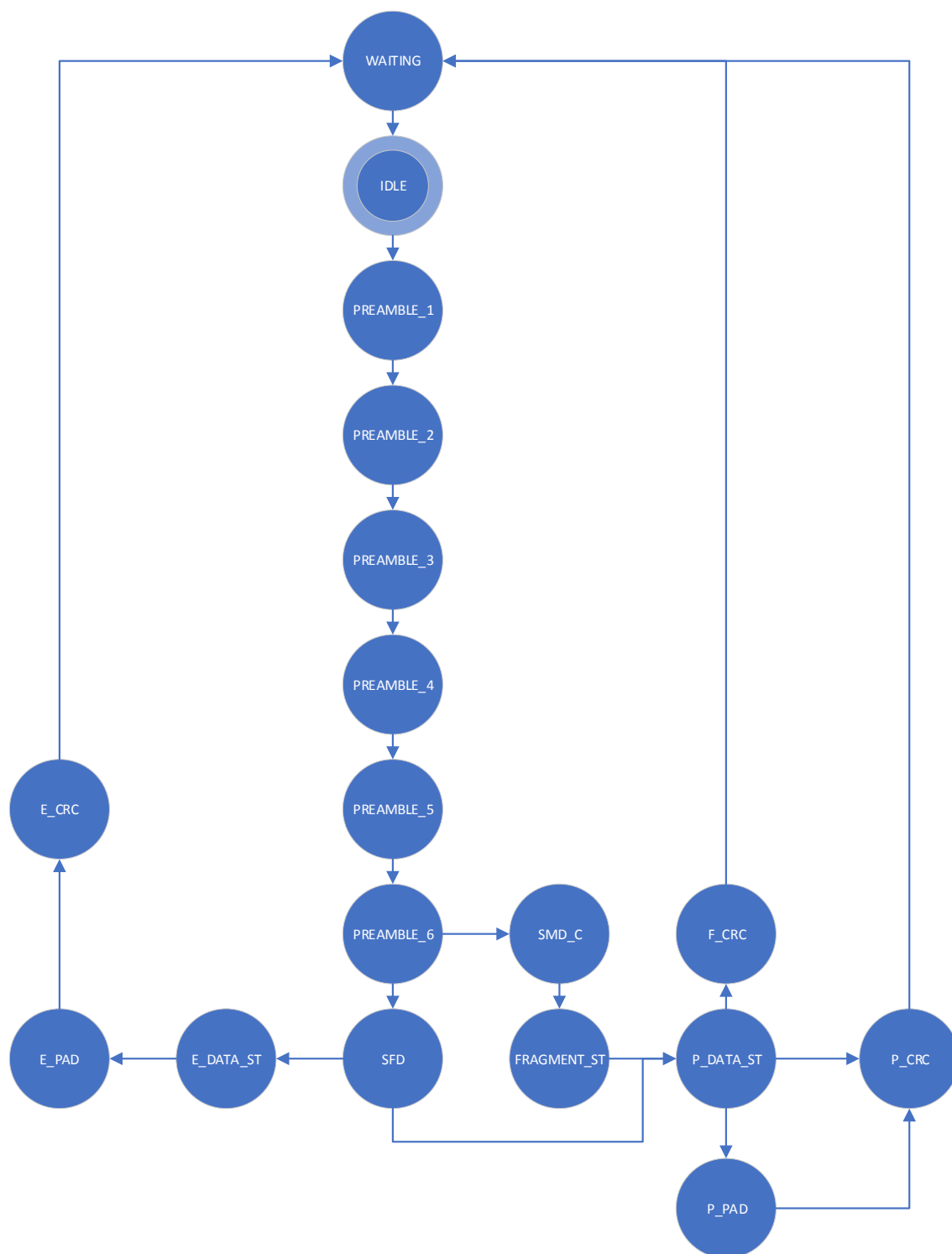
Prechody medzi stavmi sú vykreslené na obr.7.3. V tabuľke 7.3 sú vypísané výstupy z RX MACu pri danom stave. V tabuľke 7.4 sú popísané prechody medzi stavmi a podmienky, ktoré musia byť dodržané.

Tabuľka 7-3: Tabuľka prechodov medzi stavmi pre automat v TX MAC.

Stav	Nasledujúci stav	podmienka
idle	preamble_1	e_sop or e_long or p_sop or p_long or fragment_cnt > 0
preamble_1	preamble_2	
preamble_2	preamble_3	
preamble_3	preamble_4	
preamble_4	preamble_5	
preamble_5	preamble_6	transmission = 11
	smd_c	transmission /= 11
preamble_6	sfd	
sfd	e_data_st	transmission = 01
	p_data_st	transmission = 10
smd_c	fragment_st	
fragment_st	p_data_st	
e_data_st	e_pad	e_eop and e_data_count < 64
	e_crc	e_eop and e_data_count > 63
p_data_st	p_pad	p_eop = 1 and p_data_count < 64
	f_crc	e_sop and data_count > 63 and (p_data_count + 64) < pkt_size
	p_crc	p_eop = 1 and p_data_count > 63 or p_data_count = MAX_FRAME_DATA_BYTES
e_pad	e_crc	e_data_count > 63
p_pad	p_crc	p_data_count > 63
e_crc	waiting	crc_count = 3
f_crc	waiting	crc_count = 3
p_crc	waiting	crc_count = 3
waiting	idle	wait_count = 12

Tabuľka 7-4: Tabuľka výstupov pre stavový automat v TX MAC.

Stav	data			valid	eop	sop
	transmission=10	transmission=01	transmission=11			
idle	0x55	0x55	0x55	1	0	1
preamble_1	0x55	0x55	0x55	1	0	0
preamble_2	0x55	0x55	0x55	1	0	0
preamble_3	0x55	0x55	0x55	1	0	0
preamble_4	0x55	0x55	0x55	1	0	0
preamble_5	0x55	0x55	0x55	1	0	0
preamble_6	0x55	0x55	0x00	1	0	0
sfd	0xD5	SMD-Ix	0x00	1	0	0
smd_c	0x00	0x00	SMD-Cx	1	0	0
fragment_st	0x00	0x00	FRG_cnt	1	0	0
e_data_st	e_data	0x00	0x00	1	0	0
p_data_st	0x00	p_data	p_data	1	0	0
e_pad	0x00	0x00	0x00	1	0	0
p_pad	0x00	0x00	0x00	1	0	0
e_crc	exp_crc	0x00	0x00	1	if(crc=3) 1	0
f_crc	0x00	frag_crc	frag_crc	1	if(crc=3) 1	0
p_crc	0x00	preemp_crc	preemp_crc	1	if(crc=3) 1	0
waiting	0x00	0x00	0x00	1	0	0



Obr. 7.3: Diagram stavového automatu v TX MAC.

7.3 CRC_REMOVER

Tento komponent odstraňuje posledné štyri bajty z dát obdržaných na príjmovej strane a preposiela všetky signály zahrnuté v rozhraní Avalon ST smerom k hostovi, signál eop, z Avalon ST rozhrania, príde k hostovi o štyri takty skôr ako by mal, to je spôsobené tým, že sa posledné štyri bajty odstránili a tým je skôr koniec vysielaných dát. Podobne je to s potvrdzovacím signálom valid, toho trvanie sa skrúti tiež o štyri takty z tých istých dôvodov.

7.4 Syntéza

Pre správne fungovanie dizajnu je potrebné aby minimálna frekvencia dizajnu bola vyššia ako 125 MHz. Znížením by sa musela znížiť prenosová rýchlosť, tá bola stanovená na 1 Gb/s. Pre syntézu s týmto dizajnom vyšlo, že minimálna frekvencia tohto dizajnu pre FPGA z rodiny STRATIX IV, konkrétne zariadenie EP4SGX70HF35C2, pre pracovnú teplotu 85 °C 161,55 MHz a pre pracovnú teplotu 0 °C je maximálna frekvencia 168,35 MHz.

Ďalej zo syntézy vyšlo, že pre zvolené FPGA dizajn spotrebuje 1400 z 58080 kombinačný ALUT tabuliek, čo je využitie na 2 % a ešte spotrebuje 973 z 58080 logických registrov. Výslednky a syntézy a časovej analýzy sú zhrnuté v tabuľke 7-5.

Tabuľka 7-5: Prehľad výsledkov syntézy.

Maximálna pracovná frekvencia pri 85 °C	161,55 MHz
Maximálna pracovná frekvencia pri 0 °C	168,35 MHz
ALUT	1400
Logické registre	973

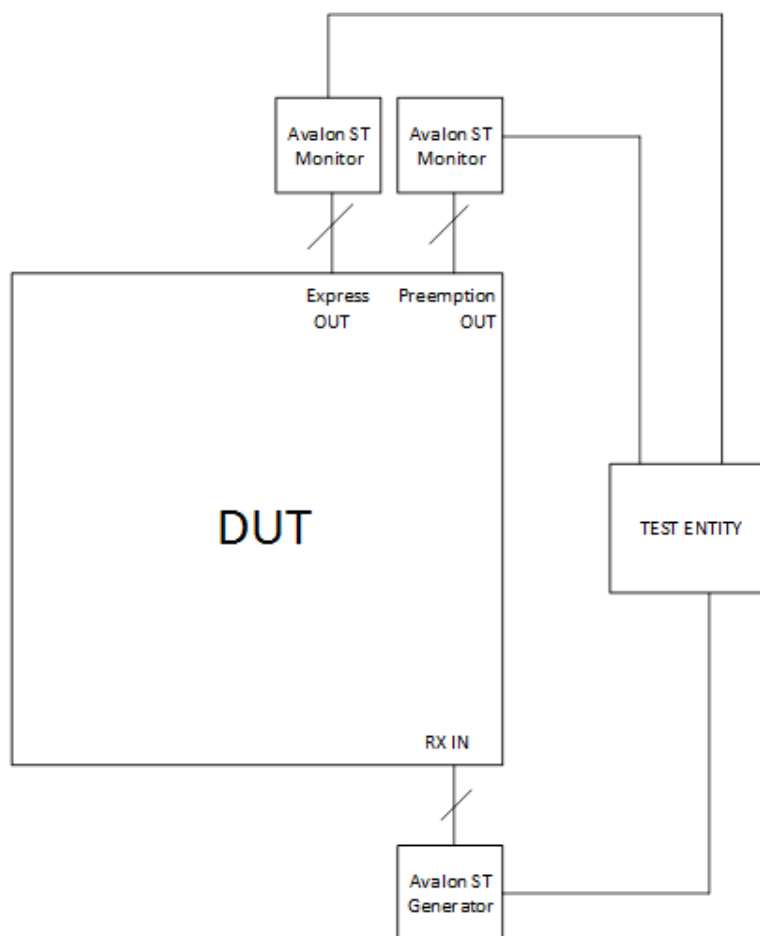
Pridat zhrnujúcu tabuľku

8 SIMULÁCIE

Pre odsimulovanie a otestovanie dizajnu boli použité dva testbenche. Prvý `tb_top_rx` testuje a simuluje iba RX stranu finálneho dizajnu. Naopak testbench `tb_top_final` testuje celý dizajn, kde sa RX strana používa pre kontrolu formátu odoslaných rámcov z TX strany.

8.1 Test bench `tb_top_rx`

V tomto testbenchi sú všetky výstupy z TX strany zadefinované ako open, čiže nie sú nikde pripojené, na vstupy TX strany sú pripojené logické jednotky. Na vstup RX strany je pripojený generátor rámcov s rozhraním Avalon ST, to isté ako je použité v celom dizajne, až na výstup z RX strany. Na výstup z RX strany sú pripojené dva monitory, ktoré zachytávajú a porovnávajú rámce. Tieto monitory majú tiež Avalon ST rozhranie, ale rozšírené o Error signál, presne tak ako na výstupe RX strany.



Obr. 8.1: Bloková schéma test bencu `tb_top_rx`.

V tomto testbenchy sa vykonávajú štyri nezávislé test procedúry. Ktoré sú popísané v nasledujúcich podkapitolách. Tieto test procedúry sa snažia čo najviac pokryť všetky funkcie dizajnu RX strany a overiť jej správnu funkčnosť. Na Obr. 8.1 je znázornené zapojenie testovaného dizajnu a verifikačných BFM.

8.1.1 Test procedúra tp_preamble

Táto test procedúra sa testuje, že žiadne prichádzajúce rámce do RX strany, s nesprávnou preambulou, prípadne s nesprávnym SFD, SMD alebo poradím fragmentu nemôžu prejsť ďalej cez RX stranu do monitoru, pre porovnanie správnosti sa použili rámce s korektnou preambulou a korektným SFD, SMD a poradím fragmentov.

Testovanie prebiehalo následne, najskôr sa posielali rámce s chybnou časťou preamble, od chyby preamble v prvom bajte až po chybu preamble v poslednom bajte, siedmom. Následne sa spravila chyba v ôsmom bajte rámcu, nastavila sa tam hodnota ktorá nie je zhodná s hodnotou SFD a zároveň nie je zhodná s hodnotami SMD-Cx. Po odoslaní týchto rámcov sa odoslal referenčný rámec s korektnými bajtmi v preamble a so správnou hodnotou SFD Dĺžka dátovej časti rámcu bola 64 bajtov. Dátová časť rámcu bola vygenerovaná pre ľahšiu kontrolu inkrementáciou iniciálnej hodnoty, v tomto prípade 0x11. Na koniec dátovej časti bolo pripojené. Štôr bajtové pole s korektným FCS, ktoré sa počítalo pomocou funkcie, ktorá bola vygenerovaná zo zdroju na internete.

Z výsledkov simulácie vyšlo, že žiadny rámec alebo fragment rámcu s chybnou preambulou, SFD, SMD alebo poradím fragmentu, neprejde cez RX stranu ďalej do hosta. Rámce so správnymi inicializačnými poľami sa začnú spracovávať a v prípade ich správnej veľkosti a správnej výslednej hodnoty po cyklickej redundantnej kontrole sa ďalej posielajú ako validné.

8.1.2 Test procedúra tp_express

Cieľom tejto test procedúry je overiť všetky možnosti na RX strane ktoré môžu vzniknúť pri prijímaní expresných rámcov. RX strana by mala zareagovať na expresné rámce s malou dĺžkou, malou dĺžkou a chybným FCS, veľkou dĺžkou, veľkou dĺžkou a chybným FCS a dĺžkou s povoleným rozsahom ale chybným FCS, tak že ich vyhodnotí ako chybné a na koniec rámcu nastaví error signál do logickej úrovne jedna, v prípade dlhých rámcov sa po presiahnutí povolenej dĺžky nastaví error signál tiež do jednotky a prestanú sa dáta z rámcu ďalej posilať.

Testovanie prebiehalo následne, pre všetky posielané rámce sa poslala korektná preambula a SFD, dátová časť sa generovala inkrementáciou iniciálnej hodnoty, tá bola vo všetkých prípadoch nastavená na hodnotu 0x11. V prípade, že malo byť na konci správne FCS sa na koniec pripojila štvor bajtové pole, spočítané pomocou funkcie vygenerovanej na internete. Najskôr sa poslali dva rámce s malou dĺžkou, 20 bajtov, jeden s pridaným poľom so správnym FCS, čiže tento rámec mal celkovú dĺžku 24 bajtov plus preambula a SFD. Dáta očakávané na výstupe boli nastavené v monitore s dĺžkou 16 bajtov, pre rámec bez FCS, a s dĺžkou 20 bajtov pre rámec s platným FCS. Potom boli poslané dva rámce s dĺžkou 200 bajtov, jeden s poľom obsahujúcim FCS a druhý, bez tohto poľa. Pre kontrolu rámcu s platným FCS sa nastavili dáta s dĺžkou 200 bajtov a pre rámec bez FCS poľa boli nastavené dáta s dĺžkou 196 bajtov. Nakoniec sa poslali dva rámce s dĺžkou 2000 bajtov, maximálna dĺžka dátovej časti je 1520 bajtov. K prvému sa

pridalo pole s platným FCS a k druhému sa nič nepridalo. Pre odkontrolovanie oboch rámcov sa nastavili pre oba rámce dáta s dĺžkou 1520 bajtov, keďže RX strana nič nerobí s dátami ak počet bajtov presiahne maximálnu dĺžku rámcu.

Z výsledkov simulácie vyšlo, že rámce s neplatnou dĺžkou a neplatným FCS sú vyhodnotené ako chybové a posielajú sa ďalej s chybovou hláškou, pre informovanie vyššej vrstvy, to isté platí aj pre rámce s nesprávnou dĺžkou.

8.1.3 Test procedúra `tp_no_preemption`

Cieľom tejto test procedúry je overiť všetky možnosti na RX strane ktoré môžu vzniknúť pri prijímaní preempčných rámcov. RX strana by mala zareagovať na preempčné rámce s malou dĺžkou, malou dĺžkou a chybným FCS, veľkou dĺžkou, veľkou dĺžkou a chybným FCS a dĺžkou s povoleným rozsahom ale chybným FCS, tak že ich vyhodnotí ako chybné a na koniec rámcu nastaví error signál do logickej úrovne jedna, v prípade dlhých rámcov sa po presiahnutí povolenej dĺžky nastaví error signál tiež do jednotky a prestanú sa dáta z rámcu ďalej posielat'. Obťažnejšie je to s SMD, ten sa mení pri každom rámci, celkovo sa striedajú štyri hodnoty, ktoré sa cyklia po sebe.

Pre správne otestovanie sa muselo pri teste poslať päť rámcov po sebe, pre overenie cyklenia hodnôt SMD, z každej testovanej kategórii. Pre všetky rámce sa generuje preambula spolu so správnym SMD. Dátová časť pre všetky rámce je generovaná inkrementáciou iniciálnej hodnoty. Na začiatok bolo poslaných päť rámcov z dlhých 200 bajtov, preto na výstupe monitor očakáva päť rámcov po 200 bajtoch. Ďalej bolo poslaných päť rámcov po 200 bajtoch, ale na konci nebolo pridané FCS, čo spôsobí odstrihnutie posledných štyroch bajtov, takže na výstupe monitor očakáva päť rámcov s dĺžkou 196 bajtov. Pre správne dodržanie SMD prvý rámec sa posielal nie s nultou hodnotou SMD ale už s prvou. Následne sa pošle päť rámcov dlhých 2000 bajtov bez FCS a potom päť bajtov s platným FCS, pri všetkých týchto rámcoch sedí SMD. Pre testovanie oboch týchto sérií sa na monitor nastaví rámce s 1520 bajtov na porovnanie s odoslanými dátami. Nakoniec sa pošle päť rámcov s dĺžkou 20 bajtov a platným FCS a päť rámcov s dĺžkou 20 bajtov a neplatným FCS, pre všetky rámce sa vygeneruje správne SMD. Pre monitor na výstupe sa nastaví 20 bajtov na porovnanie s rámcami ktoré obsahujú FCS. Pre rámce ktoré neobsahujú FCS sa nastaví na monitor 16 bajtov.

Z výsledkov simulácie vyšlo, že rámce s neplatnou dĺžkou a neplatným FCS sú vyhodnotené ako chybové a posielajú sa ďalej s chybovou hláškou, pre informovanie vyššej vrstvy, to isté platí aj pre rámce s nesprávnou dĺžkou. Ďalej sa podarilo overiť, že cyklenie SMD je potrebné pre posielanie rámcov na RX stranu.

8.1.4 Test procedúra `tp_preemption`

Cieľom tejto test procedúry je overiť všetky možnosti na RX strane ktoré môžu vzniknúť pri prijímaní fragmentov preempčných rámcov. RX strana by mala zareagovať na preempčné rámce s malou dĺžkou, malou dĺžkou a chybným FCS, veľkou dĺžkou, veľkou dĺžkou a chybným FCS a dĺžkou s povoleným rozsahom ale chybným FCS, tak že ich vyhodnotí ako chybné a na koniec rámcu nastaví error signál do logickej úrovne jedna, v prípade dlhých rámcov sa po presiahnutí povolenej dĺžky nastaví error signál tiež do jednotky a prestanú sa dáta z rámcu ďalej posielat'. Obťažnejšie je to s SMD, ten sa mení pri každom rámci, celkovo sa striedajú štyri hodnoty, ktoré sa cyklia po sebe.

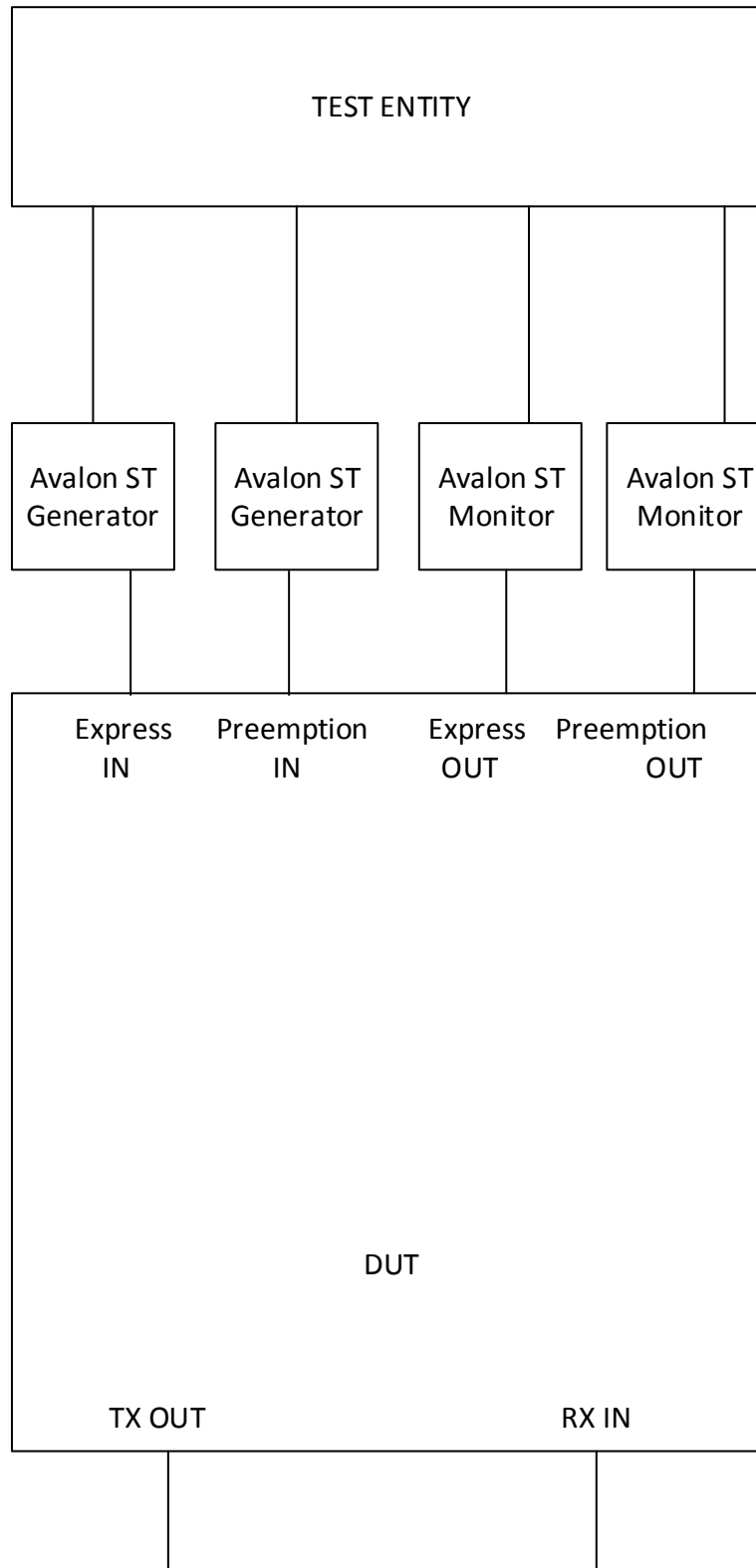
Pri tejto test procedúre sa na začiatok poslalo šesť rámcov s dĺžkou 700 bajtov, ktoré boli rozdelené na sedem fragmentov po 100 bajtoch, s dodržaním správneho formátu a s korektnými poľami obsahujúcimi FCS. Následne sa pošle jeden rámec dlhý 700 bajtov. Ten je rozdelený na sedem fragmentov, všetky fragmenty obsahujú chybné pole s FCS až na iniciálny fragment, ten obsahuje korektné pole s FCS. Potom bol opäť poslaný rámec s dĺžkou 700 bajtov, rozdelený na sedem fragmentov po 100 bajtoch. Všetky fragmenty obsahujú pole so správnym FCS, ale obsahujú nesprávne poradie fragmentu, čo znamená, že na vyššiu vrstvu sa dostane len iniciálny fragment. Potom sa poslali dva rámce s dĺžkou 10 bajtov, prvý obsahoval pole so správnym FCS a ten druhý nie. To znamená, že z prvého rámcu by mali prejsť všetky dátové bajty s propagovaným error signálom. Z druhého rámcu by malo prejsť len prvých šesť bajtov, keďže posledné štyri by mali byť odstrihnuté, kvôli neprítomnosti FCS poľa. Nakoniec sa poslali dva 7000 bajtové rámce, nespĺňajú rozmery štandardného rámcu. Obidva sú rozdelené na fragmenty s dĺžkou 1000 bajtov. Prvý fragment z rámcov by mal prísť bez problémov, ale z ostatných fragmentov prejdú len dáta po 520. bajt.

Pomocou tejto test procedúry sa podarilo overiť fungovanie skladania fragmentov do celých rámcov na RX strane. Ku kompletnému poskladaniu rámcov dôjde, jedine v prípade, že bol dodržaný správny formát rámcu. V inom prípade sa do vyššej vrstvy propaguje chybová hláška v tvare error signálu.

8.2 Test bench tb_top_final

V tomto testbenchi je výstup z TX strany napojený na vstup RX strany. Na obidva vstupy pre TX stranu sú napojené Avalon ST generátory. Na výstup z RX strany sú pripojené dva monitory, ktoré zachytávajú a porovnávajú rámce. Tieto monitory majú tiež Avalon ST rozhranie, ale rozšírené o Error signál, presne tak ako na výstupe RX strany.

V tomto testbenchy sa vykonávajú tri nezávislé test procedúry. Ktoré sú popísané v nasledujúcich podkapitolách. Tieto test procedúry sa snažia čo najviac pokryť všetky funkcie dizajnu TX strany a overiť jej správnu funkčnosť, keďže RX strana bola overená v test benchy tb_top_final a môže sa použiť ako otestovaný príjmač. Na Obr. 8.2 je znázornené zapojenie testovaného dizajnu a verifikačných BFM.



Obr. 8.2: Bloková schéma test bencu `tb_top_final`.

8.2.1 Test procedúra tp_padding

Cieľom tejto test procedúry je overiť, že pri krátkom dátovom pakete vygeneruje TX strana rámce s minimálnou dĺžkou, kde zvyšné dátové pole bude vyplnené bajtami s hodnotou 0x00.

Pre testovanie sa vygenerovali rámce s dĺžkou 10 bajtov. Ako očakávané rámce na monitoroch sa nastavili rámce s dĺžkou 60 bajtov, kde len prvých 10 bajtov bolo generovaných z počiatočnej hodnoty 0x11 za pomoci inkrementácie. Zvyšné bajty v očakávanom rámci boli nastavené na hodnotu 0x00.

Z výsledkov tejto test procedúry vyplýva, že TX strana zvláda funkciu paddingu a pri rámcoch, ktoré sú kratšie ako je minimálna dĺžka rámcu, ich dokáže na minimálnu dĺžku predĺžiť za pomoci pridávania bajtov s hodnotou 0x00.

8.2.2 Test procedúra tp_back_to_back

Cieľom tejto test procedúry je overiť funkcie pre TX stranu, kde sa overia funkcie pre preempciu rámcov pri vysielaní. Po príchode expresných dát sa pozastaví vysielanie preempčného rámcu a po dokončení tohto vysielania sa pokračuje s vysielaním preempčného rámcu.

Počas test procedúry sa poslalo 7 rámcov s dĺžkou 1000 bajtov. Časový odstup medzi dátami prichádzajúcimi do preempčného vstupu pre TX MAC bol nastavený na 1 μ s. Popri tom sa paralelne k odoslaniu nastavilo 12 rámcov s dĺžkou 64 bajtov. Časový odstup medzi dátami prichádzajúcimi do expresného vstupu pre TX MAC bol nastavený na 20 μ s. Na monitory sa nastavili výstupné hodnoty, ktoré sú zhodné so vstupnými dátami.

Z výsledkov pre túto test procedúru možno potvrdiť správnu funkčnosť vysielacej časti dizajnu. Tá dokáže pozastaviť vysielanie preempčných rámcov a vo vysielaní uprednostniť expresné. Po ich odvysielaní dokáže obnoviť vysielanie preempčného rámcu.

8.2.3 Test procedúra tp_long_pkt

V tejto test procedúre sa overuje nadštandardná funkcia v TX strane. Oba typy rámcov (preempčné aj expresné), ktoré prídu na vstup TX strany a sú dlhšie ako je maximálna povolená dĺžka. TX vrstva dokáže tieto rámce rozdeliť pred odoslaním na rámce s maximálnou dĺžkou. Ďalej sa v tejto test procedúre overuje správnosť preempcie pri týchto extra dlhých paketoch.

V tejto test procedúre sa najskôr zo začiatku nastaví vysielanie preempčného rámcu s dĺžkou 1600 bajtov. Vysielanie tohto rámcu sa preruší vysielaním expresného rámcu, ktorý má tiež dĺžku 1600 bajtov. Na monitory sa nastaví referenčné dáta na porovnanie. Pre každý z týchto rámcov, ale signály zo zbernice ako SOP a EOP boli nastavené tak aby ohraničili vzniknuté rámce po rozdelení TX stranou.

Táto test procedúra overila, že rámce dlhšie ako je maximálna povolená dĺžka sú pred vysielaním rozdelené na kratšie rámce s povolenou dĺžkou. Ďalej tu bola overená funkčnosť preempcie pri týchto nadštandardne dlhých rámcoch.

8.3 Výsledky test procedúr

Pomocou test procedúr boli overené všetky testované funkcie dizajnu. Všetky funkcie pracujú presne tak ako boli navrhnuté. Výpisy z test procedúr sú priložené na CD, ktoré je priložené k diplomovej práci spolu s testovaným dizajnom, verifikačnými BFM, test benchmi a test procedúrami. Spúšťanie test procedúr je popísané na CD nosiči v súbore readme.txt.

9 ZÁVER

Cieľom tejto diplomovej práce bolo preštudovať si podštandardy, zo štandardu IEEE 802.1Q, z ktorých sa jeden zvolil na implementáciu. Na preštudovanie sa vybrali podštandardy 802.1QBV a 802.1QBU, ktoré sú popísané v kapitolách 2 až 4. Ďalej boli spočítané ich teoretické vlastnosti ako priepustnosť a latencia časovo kritických rámcov. Tieto vlastnosti boli porovnané medzi sebou a štandardným ethernetom. Z teoretických výsledkov vyšlo, že 802.1QBU s minimálnou dĺžkou fragmentu 64 bajtov je v pomere priepustnosť a latencia pre expresný rámec najvhodnejší. Avšak v aplikáciách pre sieť kde nie je tolerovaná latencia pre časovo kritické rámce sa 802.1QBU nemôže ako samotné použiť. Pre tieto siete je možné, z uvažovaných štandardov použiť 802.1QBV alebo kombináciu 802.1QBU spolu s 802.1QBV. Z týchto dvoch možností vychádza priepustnosť najlepšie pre kombináciu 802.1QBU a 802.1QBV, konkrétne možnosť s použitím minimálnej dĺžky fragmentov 64 bajtov.

Ďalším cieľom tejto diplomovej práce bolo vytvoriť dizajn funkčného IP jadra pre štandard IEEE 802.1QBU. To bolo vybrané z dôvodu nižšej náročnosti na použité algoritmy. Formát a kódovanie rámcov, ktoré bolo použité, sa zvolilo definované v štandarde 802.3BR. Pre prípadnú implementáciu do FPGA, od Intel FPGA (dakedy Altera), a lepšiu komunikáciu s možnými IP blokmi vygenerovanými pre FPGA od zvolenej spoločnosti, sa zvolilo rozhranie Avalon ST, ktoré je plne podporované výrobcom. Funkcie vytvoreného IP jadra je popísané v kapitole 7.

Pre verifikáciu sa použili dva test benche v ktorých boli použité verifikačné BFM pre rozhranie Avalon ST. V prvom test benchy sa overovali požadované funkcie RX strany vytvoreného dizajnu, generátorom transakcií na vstupe RX strany a dvomi monitormi na expresnom a preempčnom výstupe RX strany. V druhom test benchy bolo zavedená priama komunikácia medzi TX a RX stranou, na vstupy pre TX stranu sa pripojili Avalon ST generátory a na výstup z RX strany sa pripojili Avalon ST monitory, navyše do test benchu boli pridané asserty, ktoré hlásia chybu pri nepovolenej dĺžke rámcu alebo pri kratšej vzdialenosti medzi rámcami ako je povolená. Test benche boli riadené test procedúrami, každá z test procedúr sa zameriavala na overenie inej funkcie navrhnutého dizajnu. Všetky funkcie dizajnu (podľa zoznamu v kapitole sedem odstavce dva a tri) boli buď otestované a v prípade identifikovanej chyby doladené v dizajne.

Pre overenie syntetizovateľnosti dizajnu bol použitý program Quartus Prime od Intel Fpga. V tomto programe úspešne bola vykonaná syntéza dizajnu pre FPGA Stratix IV (EP4SGX70HF35C2), z ktorej vyšlo, že navrhovaný dizajn zaberie približne 3 % z dostupných zdrojov, presné čísla sú uvedené v kapitole 7.3. Z časovej analýzy vyšlo, že dizajn bude môcť fungovať na frekvencii 161,55 až 168,35 MHz, čo je viac ako stanovená frekvencia 125 MHz, pre prenosovú rýchlosť 1 Gb/s.

Podľa výsledkov zo simulácií sa overila funkčnosť navrhovanej logiky a správnosť výsledku. Tento dizajn je možné syntetizovať a nahrat' do FPGA, pretože výsledky syntézy a časovej analýzy potvrdili, že navrhnutý dizajn dokáže splniť priestorové a časové požiadavky daného FPGA (pri zachovaní 97 % zvyšného FPGA).

Zo zistených údajov sa poskytuje ako pokračovanie tejto práce, ju rozšíriť o štandard 802.1QBV, vďaka čomu by sa toto IP jadro mohlo aplikovať v časovo citlivých sieťach. Ďalej je odporúčené prácu rozšíriť o rozhranie, ktoré je schopné prepojiť na fyzickú

vrstvu, napríklad GMII. Ošetrenie pretečenia pomocou FIFO bufferov, ktoré by mali ideálne pracovať na dvoch clock doménach, z dôvodu možnosti použitia dizajnu vyššej vrstvy v inej clock doméne. Ďalej je tiež doporučené, rozšírenie error signál na viac bitov, pre možnosť špecifikácie vzniknutej chyby.

LITERATÚRA

1. **Simonea, Paul.** [www.globalknowledge.com](http://ru6.cti.gr/bouras-old/WP_Simoneau_OSIModel.pdf). [Online] 11 October 2017. http://ru6.cti.gr/bouras-old/WP_Simoneau_OSIModel.pdf. ISSN: 0974-5572.
2. **Digital Equipment Corporation, Intel Corporation, Xerox Corporation.** *The Ethernet A Local Area Network Data Link Layer and Physical Layer Specifications*. 1982. AA-K759B-TK.
3. **Franklin, Michael Gregg Stephen Watkins George Mays Chris Ries Ronald Bandes Brandon.** *Hack the Stack*. Rockland : Syngress Media,U.S., 2006. 9780080507743.
4. **IEEE.** *.802.2 Part 2: Logical Link Control*. New York : IEEE, 1998. 1-55937-959-6.
5. **IEEE.** *IEEE Standard for Ethernet*. New York : IEEE, 2012. ISBN 973-07381-7312-2.
6. **IEEE.** <http://www.ieee802.org>. <http://www.ieee802.org/1/pages/802.1bv.html>. [Online] IEEE 802.1, 10 November 2017.
7. **IEEE.** *IEEE P802.1Qbv/D3.1*. New York : IEEE, 2015.
8. **IEEE.** <http://www.ieee802.org>. <http://www.ieee802.org/1/pages/802.1bu.html>. [Online] IEEE, 15 November 2017.
9. **IEEE.** *IEEE P802.1Qbu/D3.0*. New York : IEEE, 2015.
10. **IEEE.** *IEEE P802.1Qbu/D0.0*. New York : IEEE, 2013.
11. **IEEE.** <http://www.ieee802.org>. http://www.ieee802.org/3/br/Baseline/8023-IET-TF-1405_Winkel-iet-Baseline-r4.pdf. [Online] 10 December 2017.
12. **Boiger, Christian.** <http://www.ieee802.org>. <http://www.ieee802.org/1/files/public/docs2014/bu-boiger-CBS-Preemption-Support-0914-v01.pdf>. [Online] 3 November 2017. [Cited: 10 September 10.]
13. Avalon ® Interface Specifications. In: <https://www.altera.com> [online]. 2018 [cit. 2018-05-21]. Dostupné z: https://www.altera.com/en_US/pdfs/literature/manual/mnl_avalon_spec.pdf
14. *Transmission Efficiency (Data Communications and Networking)* [online]. [cit. 2018-05-21]. Dostupné z: <http://what-when-how.com/data-communications-and-networking/transmission-efficiency-data-communications-and-networking/>